

Aaron Andal : Backup

CREATE BACKUP

PG_DUMP - Copy 1 database

- **pg_dump -v -d database -f file.sql** → Copia de seguridad de toda la base de datos tipo SQL.
- **pg_dump -v -F d training -f trainingdumpdir** → Copia de seguridad en formato **DIRECTORIO**.
- **pg_dump database -t comandes -f file.sql** → Copia de seguridad de la tabla comandas tipo SQL
- **pg_dump -d database -F d training -f training_dir** → **Copia de seguridad de toda la base de datos pero en FORMATO DIRECTORIO**
- **pg_dump -d database -F t -f training.tar** → **Copia de seguridad del directorio en formato TAR.**
- **pg_dump training2 -F t | gzip > training2.tgz** → **Fichero TAR muy grande.**
- **gunzip -c training2.tgz | pg_restore -d training** → **Restaurar TAR muy grande.**
- **pg_dump -F t training -j 5 | gzip > training_bkp4.tar.gz** → **NO PODEMOS HACERLA PORQUE SOLO SOPORTA EL FORMATO -d de directorios**
- **pg_dump --dbname=training --file=training_bkp4.sql**
- **pg_dump --dbname=training -t rep_vendes --username=ix36579183 --format=custom --file=training_bkp6.bin** → Backup lógico sólo de la tabla **rep_vendes**
- **pg_dump --dbname=training -t comandes --username=ix36579183 --format=plain --file=training_bkp5.sql** → Backup en formato

EXAMPLES

To dump a database called mydb into a SQL-script file:

```
$ pg_dump mydb > db.sql
```

To reload such a script into a (freshly created) database named newdb:

```
$ psql -d newdb -f db.sql
```

To dump a database into a custom-format archive file:

```
$ pg_dump -Fc mydb > db.dump
```

To dump a database into a directory-format archive:

```
$ pg_dump -Fd mydb -f dumpdir
```

To dump a database into a directory-format archive in parallel with 5 worker jobs:

```
$ pg_dump -Fd mydb -j 5 -f dumpdir
```

To reload an archive file into a (freshly created) database named newdb:

```
$ pg_restore -d newdb db.dump
```

To reload an archive file into the same database it was dumped from, discarding the current contents of that database:

```
$ pg_restore -d postgres --clean --create db.dump
```

To dump a single table named mytab:

```
$ pg_dump -t mytab mydb > db.sql
```

To dump all tables whose names start with emp in the detroit schema, except for the table named employee_log:

```
$ pg_dump -t 'detroit.emp*' -T detroit.employee_log mydb > db.sql
```

To dump all schemas whose names start with east or west and end in gsm, excluding any schemas whose names contain the word test:

```
$ pg_dump -n 'east*gsm' -n 'west*gsm' -N '*test*' mydb > db.sql
```

The same, using regular expression notation to consolidate the switches:

```
$ pg_dump -n '(east|west)*gsm' -N '*test*' mydb > db.sql
```

To dump all database objects except for tables whose names begin with ts_:

```
$ pg_dump -T 'ts_*' mydb > db.sql
```

PG_DUMPALL - Copy ALL DATABASE

RESTORE BACKUP

PGRESTORE

PSQL →

psql --dbname=training --username=isx36579183 < training_bkp5.sql → Restaurar la tabla rep_vendes

DROP TABLE rep_vendes CASCADE;

pg_restore -d training -Ft -t rep_vendes < fitxer.tar → Restaurar solo una tabla

pg_restore -Fc -d training training_bkp6.bin → Restablecer la tabla rep_vendes en BINARIO

pg_dump -h server1 training | psql -h server2 training → Backup de Training i se restaura on the fly a server2.

pg_dump -h 192.168.0.123 -U isxxxxxx -Ft training < training192_168_0_123.tar → Backup que permite hacer una copia de seguridad tipo tar a training que está a 192.168.0.123

pg_dumpall -f backup_all_db.sql o **psql < backup_all_db.sql**

1 pg_dump -v -d database -f file.sql → Copia de seguridad de toda la base de datos tipo SQL.

2 pg_dump -F d training -f training_dir → Copia de seguridad de toda la base de datos pero en forma de Directorio

3 pg_dump training -F t -f training.tar → Formato TAR de toda la base de datos

4 pg_dump training2 -F t | gzip > training2.tgz → Backup en formato GZIP y es muy grande

5 gunzip -c training2.tgz | pg_restore -d training → Restaurar modo GZIP y es muy grande

6 pg_dump -F t training -j 5 | gzip > training_bkp4.tar.gz → NO PODEMOS HACERLA PORQUE SOLO SOPORTA EL FORMATO -d de directorios

pg_dump: error: parallel backup only supported by the directory format

7 file training_bkp4.sql → Asciii

8 pg_dump --dbname=training -t comandes --username=isx36579183 --format=plain --file=training_bkp5.sql → Copia de seguridad logica de la tabla solo COMANDAS

9 isx36579183@j05:~/Documents/m02/dataBases\$ pg_dump --dbname=training -t rep_vendes --username=isx36579183 --format=custom --file=training_bkp6.bin → Copia de seguridad sólo de la tabla rep_vendes de tipo BINARIO

```

# 10 psql --dbname=training --username=isx36579183 < training_bkp5.sql → Restablecer la base de datos
COMANDES en formato SQL
o psql -d training < comandes.sql
# 11 tar xvf training_bkp2.tar → Extraer con TAR
# 12 ls -la \*.dat restore\* → Revisar los ficheros DAT y SQL
# 13 DROP TABLE rep_vendes CASCADE;
# 14 pg_restore -d training -Ft -t rep_vendes < fitxer.tar → Revisar
# 16 pg_restore -Fc -d training training_bkp6.bin o pg_restore -d training -F c -t rep_vendes < rep_vendes.dat→
Lo mismo que el anterior pero en FORMATO BINARIO
# 17 pg_restore -C -d training training.dat→ Sense utilitzar aquesta última i amb una única ordre restaura la base
de dades `training`. SIN CREATEDB.
# 18 pg_restore -C -d template1 training2.dat → Restore
# 19 psql -d training --set ON_ERROR_STOP=on < training.sql
# 20 psql --single-transaction training < training.sql → Restauración en modo transacción o todo o no hace nada
# 21 psql -1 training < training.sql`
# 22 pg_dumpall > backup_all_db.sql → Backup de todas las bases de datos de un clúster, necesita ADMIN
# 23 psql < backup_all_db.sql → Restore de toda la base de datos de un cluster .SQL , necesita ADMIN
# 24 pg_dump -h 10.200.244.206 -U isx36579183 -F t training > training_bkp_company.tar → Copia de seguridad
tipo tar a la Base de datos Training que está en una IP.
# 25 tar tf training_bkp_company.tar → Extraer TAR
# 26 pg_dump -h server1 training | psql -h server2 training → Backup de la base de datos training de Server1 que
restaura on the fly a server2 → pg_dump -h server1 training | psql -h server2 training o pg_dump -h
10.200.244.229 training > training_bkp2_company.sql || psql training < training_bkp2_company.sql
cat training_bkp2_company.sql | less

```

BORG

BORGMATIC

```

# 1 borg init -e repoquey /path/to/repo → Inicializar repositorio - -e encryption

# 2 sudo borg key export /var/tmp/Borg $HOME/Documents/m02/key_m02 → Exportar una clave

# 3 sudo borg key import user@isx36579183:/var/tmp/Borg → Importar si fuera remoto
# 4 cd /var/tmp/ && sudo borg key export Borg guest@j06:/home/guest/Documents/key_j05

# 5 show data_directory

# 6 sudo borg create --list /var/tmp/Borg::Dijous /var/lib/postgresql/13/main /etc/postgresql/13/main → Crear
backup jueves
# 7 sudo borg list /var/tmp/Borg::Dijous → Listar
# 8 sudo borg delete /var/tmp/Borg::Dijous → Delete
# 9 RESTORE CON EL ARCHIVO
# 10 sudo rm -rf /etc/postgresql/13/main/
# sudo rm -rf /var/lib/postgresql/13/main
# 11 sudo borg extract /var/tmp/Borg::Dijous
# 12 sudo chown -R postgres:postgres /etc/postgresql/13/main
# 4 sudo chown -R postgres:postgres /etc/postgresql/13/main/
# 4 sudo systemctl start postgresql
# 4 psql template1
# 4
# 4 RESTORE DIFERENTES DIAS

```

Restauració a un dels diferents dies de Backup

+ Creem un primer backup, li direm Dilluns

Parem el servei:

...

```
sudo systemctl stop postgresql
```

...

Inicialitzem el repo: (Estem a /var/tmp/)

...

```
sudo borg init --encryption=repokey Borg
```

...

Exportem la clau per si de cas:

...

```
sudo borg key export Borg $HOME/Documents/m02/key_m02
```

...

Creem el primer backup (Dilluns)

...

```
sudo borg create Borg::Dilluns backup.sql
```

...

+ Eliminem un registre d'una taula d'una base de dades i un altre registre d'una taula d'una altra base de dades

...

```
postgres=# \c scott
scott=# BEGIN;
scott=*# DROP TABLE salgrade ;
DROP TABLE
```

```
scott=# \c training
training=*# DROP TABLE rep_vendes CASCADE;
```

...

+ Creem un segon backup, li direm Dimarts

Parem el servei

...

```
sudo systemctl stop postgresql
```

...

Creem el segon backup (Dimarts)

...

```
sudo borg create Borg::Dimarts backup.sql
```

...

+ Eliminem un registre més

...

```
DROP TABLE oficinas CASCADE;
```

...

+ Restaurem a Dimarts

Parem el servei:

```
...  
sudo systemctl stop postgresql  
...
```

Ens col·loquem a ` \$HOME/Documents/m02 ` i restaurem:

Revisar aquí ruta

```
...  
sudo borg extract $HOME/Documents/m02::Dimarts  
...
```

Engeguem el servei:

```
...  
sudo systemctl start postgresql  
...
```

Comprovem que el registre que havíem eliminat al pas anterior hi és, però els que havíem eliminat abans de la còpia de Dilluns no.

+ Restaurem a Dilluns

Repetir l'acció anterior

```
...  
    sudo systemctl stop postgresql  
...  
...  
sudo borg extract $HOME/Documents/m02::Dilluns  
...  
...  
sudo systemctl start postgresql  
...
```