Lecture 21 - Backup

Backup in PosgreSQL - https://youtu.be/c13OXubCcw4

From Amazon S3 - for download (same as youtube videos)

Backup in PosgreSQL

There are a number of kinds of database backup.

Cold Backup.

1. Shutdown database and then backup all the files. Copy the files to a new computer and then *remember* to do a restore to verify that you have not missed any.
2. Use a file-system like btrfs On Linux - to take a “snapshot” of the file system. Use the ability of PostgreSQL to go int “backup” mode just before the snapshot and then as soon as the snapshot is done take the database out of backup mode.

Dump of a “database”.

This is a one-time or run by “cron” the scheduler system. Remember that you sohould copy the backup to a different system - or maybe Amazon S3 (with encryption).

To do a single database I usually use PostgreSQL’s pg_dump utility.

This command must be run as a user with read permissions to the database you intend to back up.

Log in as the postgres user:

```
$ su - postgres
```

Dump the contents of a database to a file by running the following command. Replace yourDbName with the name of the database to be backed up.

```
$ pg_dump yourDbName > yourDbName.bak
```

The resulting backup file, yourDbName.bak, can be transferred to another host with scp.

To demonstrate restoring lost data, delete your example database and create an empty database in its place:

```
$ psql
postgre=# create database newdb;
postgre=# \q
```

Restore the database using psql:

```
$  psql newdb < yourDbName.bak
```

By default pg_dump will output in a compressed format. You can also specify options for it to output the data in tar format or in .sql statements.

```
$ pg_dump --format=t yourDbName > yourDbName.tar
```

Dumps in a format that is compatible with the ‘tar’ tape archive tool.

Incremental backup.

The primary tool that I have used is Barman. There are other tools like Amanda that also provide these kind of features.

Barman is a disaster recovery solution for PostgreSQL developed and maintained by a PostgreSQL company 2ndQuadrant.

It uses the Point-In-Time-Recovery (PITR) features that are built into the database.

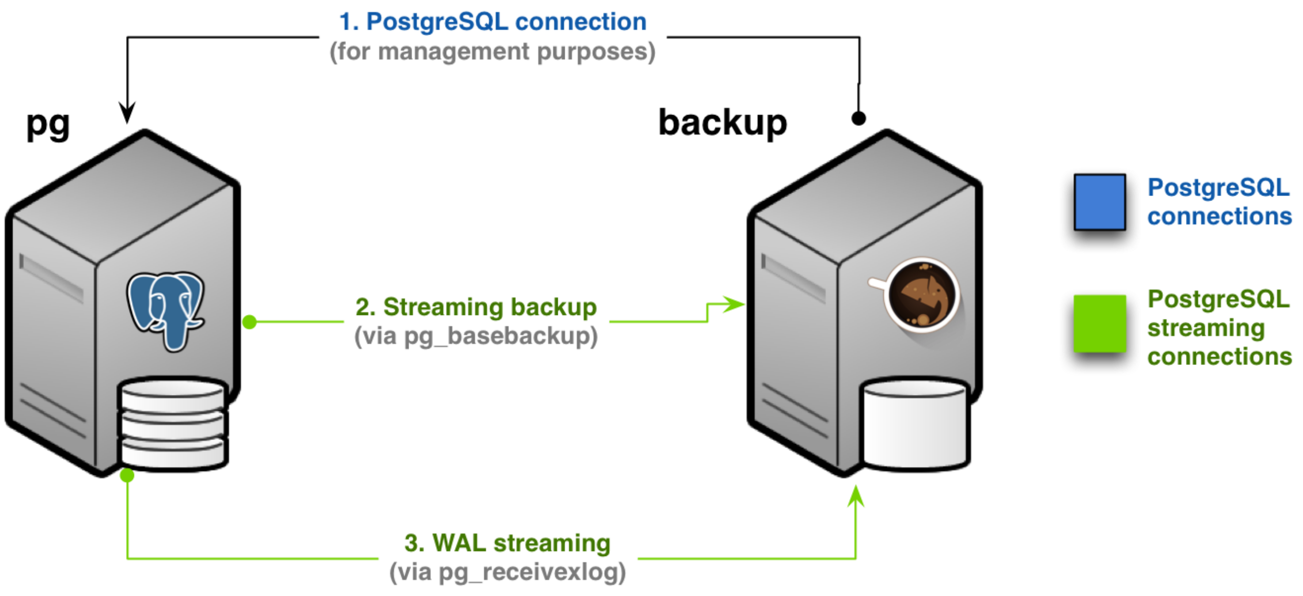
Features of Barman:

- handles multiple targets
- support for different PostgreSQL versions
- zero data loss
- streaming and/or standard archiving of WALs
- local or remote recovery
- simplified point in time recovery

Under the covers it uses a tool called ‘rsync’ for moving files. This is a really useful network backup tool that I use for all sorts of file movement - like deploying websites. Using the ‘rsync’ options in Barman you get incremental backups, parallel jobs, data deduplication, and network compression.

It is worth reading the documentation on barman, http://docs.pgbarman.org/release/2.12/

This tool is designed for a database with a backup database.



(From the barman documentation)

PostgreSQL can output all of its changes in special files called WAL - that are then transferred to a 2nd database that reads the files. It takes considerable setup to get this to work - but is definitely worth it. The “backup” database can become the primary in a few seconds - giving you a “live” backup. This system requires some configuration on the application side so that apps can switch from a primary to a backup. A new backup can then be created from either the database files or from re-running the WAL files. Barman can provide this backup “environment”.