

Set up .pgpass for PostgreSQL backups

If your backups connect over TCP or as a user other than `postgres`, `pg_dump` needs a password. Put it in a `.pgpass` file **in the home folder of the user that runs the backup**, not in the script.

Peer authentication first

If the backup runs as the `postgres` OS user over the local socket, peer authentication already works and you don't need `.pgpass` at all.

Format

One line per server, port, database and user:

```
hostname:port:database:username:password
localhost:5432:*:backup:secret
localhost:5433:*:backup:secret
```

Create it safely

Set a restrictive umask first, so the file is never briefly readable by others:

```
sudo -u postgres bash -c 'umask 077; cat >> ~/.pgpass' <<'EOF2'
localhost:5432:*:backup:secret
EOF2
sudo -u postgres chmod 600 ~/.pgpass
```

`libpq` ignores the file if its mode is wider than `0600`.

Common mistake

Writing `/root/.pgpass` while the cron job runs as `postgres` (home folder `/var/lib/pgsql`). The backup never sees that file. Check which user and home folder your job actually uses.

Test

```
sudo -u postgres psql -h localhost -p 5433 -U backup -c 'select 1' postgres
```

Revision #1

Created 24 October 2026 13:29:32 by Admin

Updated 24 October 2026 13:29:32 by Admin