



EDB Open Source Learning Day

Troubleshooting Postgres

Lætitia Avrot - Field CTO



Lætitia Avrot



- Field CTO – EDB
- Trésorière de PostgreSQL Europe
- Fondatrice de Postgres women
- Contributrice reconnue du projet PostgreSQL
- mydbanotebook.org
- psql-tips.org

Agenda

- Introduction
- Can't connect
- Can't start
- Can't replicate
- Corruption

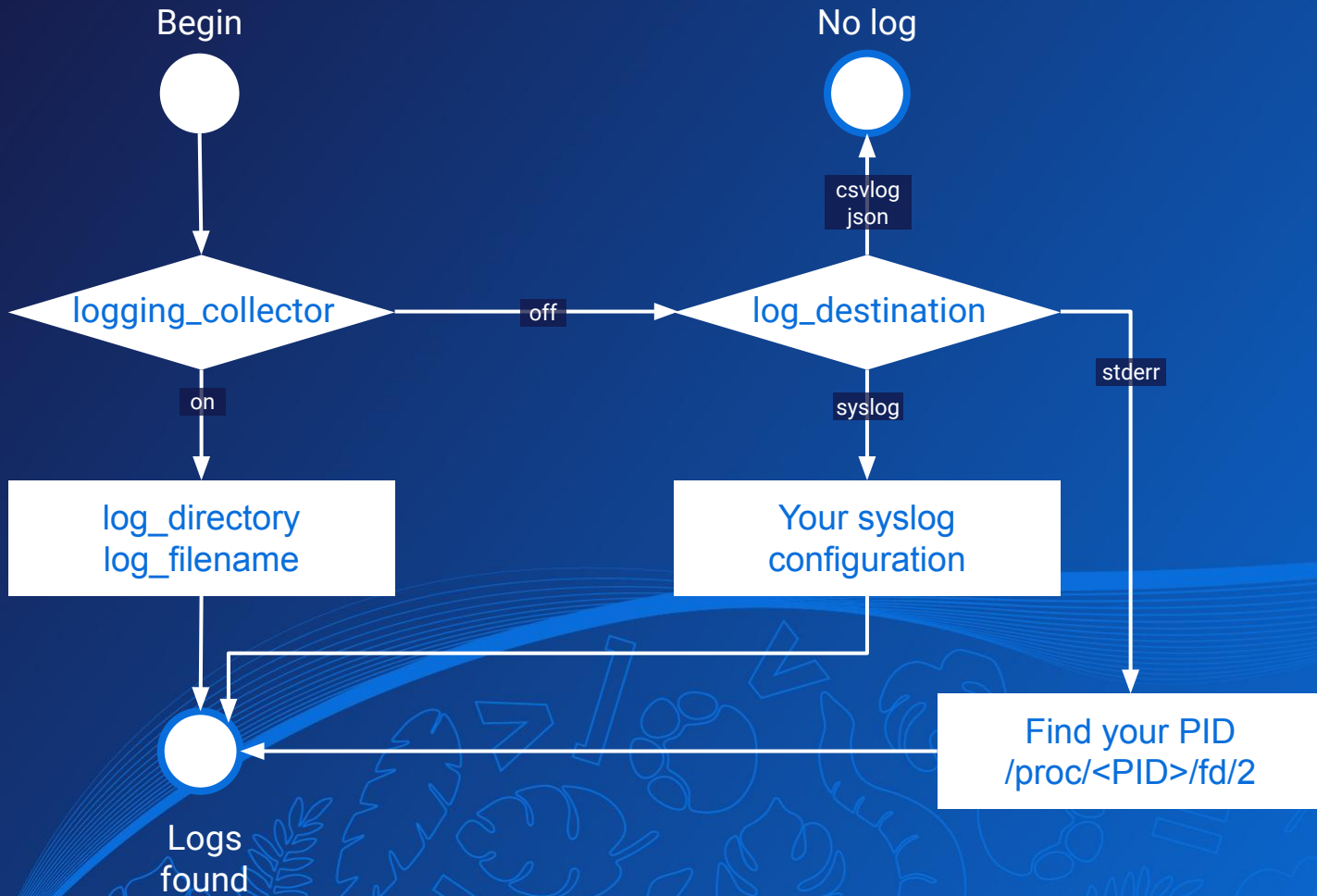
Prerequis

Il vous faudra:

- Un ordinateur
- Soit
 - Docker
 - Un shell sh
 - [Cette archive tar](#)
- Soit
 - Un moyen de se connecter en ssh à une machine AWS EC2 fournie
- Votre cerveau!

Les bases du métier

- Écouter ce que les gens disent
- Tout le monde ment
- Lire les logs



Paramétrage des logs

```
log_destination=csvlog
logging_collector=on
log_directory= whatever (not the same disk as $PGDATA)
log_filename= whatever
log_min_duration_statement=250ms (or 1s or 5s)
log_min_messages='WARNING'
log_autovacuum_min_duration=0
log_checkpoints=on
log_connections=on
log_disconnections=on
log_lock_waits=on
log_statement='ddl'
log_temp_files=0
```

Analyser les logs

- Pas en production
- Sur votre ordinateur
- (ou sur un serveur dédié)

Analyser les logs

```
create server pglog foreign data wrapper file_fdw;
```

```
create foreign table([paste definition found in doc here])  
  server pglog  
  options ([filename], format 'csv' );
```

Gardez en tête que le SQL est **Turing Complete!**

Analyser les logs

- Si vous avez le temps
 - Utilisez le SQL
- En urgence
 - Utilisez votre cerveau
 - Et des outils basiques (sed, grep, vi/emacs...)

C'est à vous de
travailler!

- Can't connect
- Can't start
- Can't replicate
- Corrupted

Conteneurs

- Dockerfile
- Construire l'image
- Démarrer le conteneur
- Régler le problème

Cheat sheet pour conteneur

Building the image	<code>docker build -t imgname .</code>
Create the container	<code>docker run -d -P --name name imgname</code>
List the containers	<code>docker ps</code>
Start/Stop the container	<code>docker stop containerid</code> <code>docker start containerid</code>
Remove the container	<code>docker rm [-f] containerid</code>
Connecting to the container	<code>docker exec -it name /bin/whatever</code>

Can't connect!

Quelqu'un vient vous voir et dit:

Je ne peux pas me connecter à Postgres. J'ai tout essayé, ça ne marche pas. Postgres est pourri, j'aurai dû choisir une autre base de données.

Que faites-vous?

Petite checklist

- ❑ Écouter la personne jusqu'au bout
- ❑ Reformuler le problème pour s'assurer que vous avez bien compris.
- ❑ Poser des questions
- ❑ Vérifier chaque fait énoncé
- ❑ Demander à prendre le contrôle, si possible

Qu'essaye de faire l'utilisateur?

```
laetitia:~/01-CantConnect|⇒ ./spin_up_postgres.sh
```

```
[...]
```

```
laetitia:~/01-CantConnect|⇒ psql -h localhost
```

```
psql: error: connection to server at "localhost" (:::1), port 5432 failed: Connection refused  
Is the server running on that host and accepting TCP/IP connections?
```

```
connection to server at "localhost" (127.0.0.1), port 5432 failed: Connection refused  
Is the server running on that host and accepting TCP/IP connections?
```


Peut-on se connecter localement?

```
laetitia:~/01-CantConnect|⇒ docker exec -it 01-cantconnect /bin/psql
```

```
psql (15.0 (Ubuntu 15.0-1.pgdg20.04+1))
```

```
Type "help" for help.
```

```
postgres=#
```

Oui

Peut-on se connecter localement en utilisant les mêmes identifiants?

```
laetitia:~/01-CantConnect|⇒ docker exec -it 01-cantconnect /bin/psql -U laetitia laetitia  
psql: error: connection to server on socket  
"/var/run/postgresql/.s.PGSQL.5432" failed: FATAL:  
Peer authentication failed for user "laetitia"
```

Non

pg_hba.conf

- Contrôle l'authentification à Postgres
- Fonctionne comme le firewall Linux

pg_hba.conf

```
laetitia:~/01-CantConnect|⇒ docker exec -it \  
01-cantconnect /bin/sh  
  
$ cat /etc/postgresql/15/main/pg_hba.conf  
  
# Database administrative login by Unix domain socket  
local all postgres peer  
# TYPE DBUSER ADDRESS METHOD  
# "local" is for Unix domain socket connections only  
local all all peer  
# IPv4 local connections:  
host all all 127.0.0.1/32 scram-sha-256  
# IPv6 local connections:  
host all all ::1/128 scram-sha-256
```

Peut-on se connecter localement en utilisant les mêmes identifiants et la même méthode de connexion?

```
laetitia:~/01-CantConnect|⇒ docker exec -it 01-cantconnect \  
  /bin/psql -h localhost -U laetitia laetitia
```

Password for user laetitia:

```
psql (15.0 (Ubuntu 15.0-1.pgdg20.04+1))  
SSL connection (protocol: TLSv1.3,  
  cipher: TLS_AES_256_GCM_SHA384, compression: off)  
Type "help" for help.  
laetitia=#
```

Oui

Essayons à nouveau depuis l'extérieur

```
laetitia:~|⇒ psql -h localhost -U laetitia laetitia
```

```
psql: error: connection to server at "localhost" (:::1),  
port 5432 failed: Connection refused  
Is the server running on that host and accepting  
TCP/IP connections?
```

```
connection to server at "localhost" (127.0.0.1),i  
port 5432 failed: Connection refused  
Is the server running on that host and accepting  
TCP/IP connections
```

Est-ce que mon port est ouvert?

```
(echo >/dev/tcp/127.0.0.1/5432) &>/dev/null && echo "open" || echo "close"
```

```
close
```

Non

C'est un conteneur!

```
laetitia:~/01-CantConnect|⇒ grep -i expose Dockerfile
```

```
# Expose the PostgreSQL port  
EXPOSE 5432
```

```
laetitia:~/01-cantConnect|⇒ cat spin_up_postgres.sh  
docker build -t cantconnect .
```

```
docker run -d -P --name 01-cantconnect cantconnect
```



Ça semble correct.

Est-ce que le port est correct?

```
laetitia:~/01-CantConnect|⇒ docker ps --format "table {{.ID}}\t{{.Image}}\t{{.Ports}}"
```

CONTAINER ID	IMAGE	PORTS
9dcad71213a0	cantconnect	0.0.0.0:55009->5432/tcp

The port is 55009!

Est-ce que le port est correct?

```
laetitia:~/01-CantConnect|⇒ psql -h localhost -p 55009 -U laetitia laetitia
```

```
psql: error: connection to server at "localhost" (:::1),  
port 55009 failed: server closed the connection unexpectedly  
This probably means the server terminated abnormally  
before or while processing the request.
```

Il faut changer ça.



Listen_addresses!

```
laetitia:~/01-CantConnect|⇒ docker exec -it 01-cantconnect /bin/sh  
  
$ grep listen_ad /etc/postgresql/15/main/postgresql.conf  
#listen_addresses = 'localhost'
```

On progresse!

Listen_addresses

- Indique sur quelle adresse IP Postgres doit écouter.
- On peut le changer pour '*'
- Nécessite un redémarrage

```
docker stop dockerId
```

```
docker start dockerId
```

Essayons à nouveau

```
laetitia:~/01-CantConnect|⇒ docker ps --format "table {{.ID}}\t{{.Image}}\t{{.Ports}}"
```

CONTAINER ID	IMAGE	PORTS
b44bcd52f08c	cantconnect	0.0.0.0:55016->5432/tcp

```
laetitia:~/01-CantConnect|⇒ psql -h 127.0.0.1 -p 55016
psql: error: connection to server at "127.0.0.1", port 55016 failed:
FATAL: no pg_hba.conf entry for host "172.17.0.1", user "laetitia", database "laetitia",
no encryption
```

On progresse (encore).

pg_hba.conf

```
laetitia:~/01-CantConnect|⇒ docker exec -it 01-cantconnect /bin/sh
```

```
$ vi /etc/postgresql/15/main/pg_hba.conf
```

```
$ cat /etc/postgresql/15/main/pg_hba.conf
```

```
# Database administrative login by Unix domain socket
```

```
local all postgres peer
```

```
# TYPE DBUSER ADDRESS METHOD
```

```
# "local" is for Unix domain socket connections only
```

```
local all all peer
```

```
# IPv4 local connections:
```

```
host all all 127.0.0.1/32 scram-sha-256
```

```
host all all 172.17.0.1/32 scram-sha-256
```

pg_hba.conf

```
$ psql
psql (15.0 (Ubuntu 15.0-1.pgdg20.04+1))
Type "help" for help.

postgres=# select pg_reload_conf();
 pg_reload_conf
-----
 t
(1 row)
```

Il ne faut pas oublier de recharger!

Essayons à nouveau

```
laetitia:~/01-CantConnect|⇒ psql -h 127.0.0.1 -p 55016 -U laetitia  
Password for user laetitia:
```

```
psql (16devel, server 15.0 (Ubuntu 15.0-1.pgdg20.04+1))  
Type "help" for help.
```

```
laetitia=#
```

Problème résolu!

Can't start!

Un de vos collègues entre dans votre bureau et dit:

Je ne sais pas ce qu'il s'est passé. Nous avons arrêté Postgres pour une opération de maintenance et il refuse de redémarrer. Je ne sais pas ce qui cloche. Nous n'avons rien changé pourtant!

Que faites-vous?

Qu'essaye de faire l'utilisateur?

```
laetitia:~/02-CantStart|⇒ docker exec -it 02-cantstart /bin/sh
```

```
$ /usr/lib/postgresql/15/bin/postgres \  
-D /var/lib/postgresql/15/main \  
-c config_file=/etc/postgresql/15/main/postgresql.conf
```

```
LOG:  skipping missing configuration file "/var/lib/postgresql/15/main/postgresql.auto.conf"
```

```
LOG:  invalid value for parameter "wal_level": "maximal"
```

```
HINT:  Available values: minimal, replica, logical.
```

```
FATAL:  configuration file
```

```
"/etc/postgresql/15/main/postgresql.conf" contains errors
```

Changer la configuration

- Ouvrir le fichier postgresql.conf
- Trouver `wal_level`
- Corriger l'erreur

C'était facile!

Essayons à nouveau

```
$ /usr/lib/postgresql/15/bin/postgres \  
-D /var/lib/postgresql/15/main \  
-c config_file=/etc/postgresql/15/main/postgresql.conf
```

```
LOG:  skipping missing configuration file "/var/lib/postgresql/15/main/postgresql.auto.conf"  
FATAL:  data directory "/var/lib/postgresql/15/main" has wrong ownership  
HINT:  The server must be started by the user that owns the data directory
```

Hein?

Les mauvaises permissions

```
$ ls -l /var/lib/postgresql/15/main
ls: cannot access '/var/lib/postgresql/15/main/pg_xact': Permission denied
ls: cannot access '/var/lib/postgresql/15/main/pg_wal': Permission denied
ls: cannot access '/var/lib/postgresql/15/main/pg_subtrans': Permission denied
ls: cannot access '/var/lib/postgresql/15/main/pg_commit_ts': Permission denied
[...]
```

```
$ whoami
postgres
```

Il faut les droits admin.

Comment devenir root?

- Pas de `sudo`
- Modification de la connexion au conteneur

```
docker stop dockerId
```

```
docker exec -it -u 0 name cmd
```

En étant root...

```
laetitia:~/02-CantStart|⇒ docker exec -it -u 0 02-cantstart /bin/sh
```

```
# ls -l /var/lib/postgresql/15
```

```
drw-rw-rw- 19 root root 4096 Oct 23 19:43 main
```

```
# ls -l /var/lib/postgresql/15/main
```

```
-rw-rw-rw- 1 root root    3 Oct 22 20:22 PG_VERSION
```

```
drw-rw-rw- 6 root root 4096 Oct 23 19:43 base
```

```
drw-rw-rw- 2 root root 4096 Oct 23 19:43 global
```

```
drw-rw-rw- 2 root root 4096 Oct 22 20:22 pg_commit_ts
```

```
# chown -R postgres:postgres /var/lib/postgresql/15/main
```

Essayons à nouveau

```
$ /usr/lib/postgresql/15/bin/postgres \  
-D /var/lib/postgresql/15/main \  
-c config_file=/etc/postgresql/15/main/postgresql.conf
```

```
LOG:  skipping missing configuration file "/var/lib/postgresql/15/main/postgresql.auto.conf"  
FATAL:  data directory "/var/lib/postgresql/15/main" has invalid  
permissions  
DETAIL:  Permissions should be u=rwx (0700) or u=rwx,g=rx (0750).
```

Sérieux?

En étant root...

```
laetitia:~/02-CantStart|⇒ docker exec -it -u 0 02-cantstart /bin/sh

# ls -l /var/lib/postgresql/15
drw-rw-rw- 19 postgres postgres 4096 Oct 23 19:43 main

# ls -l /var/lib/postgresql/15/main
-rw-rw-rw- 1 postgres postgres    3 Oct 22 20:22 PG_VERSION
drw-rw-rw- 6 postgres postgres 4096 Oct 23 19:43 base
drw-rw-rw- 2 postgres postgres 4096 Oct 23 19:43 global
drw-rw-rw- 2 postgres postgres 4096 Oct 22 20:22 pg_commit_ts

# chmod -R 0700 /var/lib/postgresql/15/main
```

Essayons à nouveau

```
$ /usr/lib/postgresql/15/bin/postgres \  
-D /var/lib/postgresql/15/main \  
-c config_file=/etc/postgresql/15/main/postgresql.conf
```

```
LOG: starting PostgreSQL 15.0 (Ubuntu 15.0-1.pgdg20.04+1) on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu  
9.4.0-1ubuntu1~20.04.1) 9.4.0, 64-bit  
LOG: listening on IPv4 address "0.0.0.0", port 5432  
LOG: listening on IPv6 address ":::", port 5432  
LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"  
LOG: database system was interrupted; last known up at 2022-10-23 20:46:37 UTC  
LOG: database system was not properly shut down; automatic recovery in progress  
LOG: invalid record length at 0/1915CC8: wanted 24, got 0  
LOG: redo is not required  
LOG: checkpoint starting: end-of-recovery immediate wait  
LOG: checkpoint complete: wrote 3 buffers (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.278  
s, sync=0.002 s, total=0.285 s; sync files=2, longest=0.001 s, average=0.001 s; distance=0 kB, estimate=0 kB  
LOG: database system is ready to accept connections
```

Bonus

Le collègue revient dans votre bureau et dit:

Il y a quelque chose de bizarre avec Postgres: nous avons pourtant bien modifié le paramètre `shared_buffer` à 128MB, mais Postgres continue à prendre plus de mémoire que ça...

Que faites-vous?

Le paramétrage de Postgres

```
postgres=# select name, setting, unit, source
postgres=# from pg_settings
postgres=# where name ~ 'shared_buffers';
name          | setting | unit | source
-----+-----+-----+-----
shared_buffers | 4194304 | 8kB  | configuration file
(1 row)

postgres=# select 4194304*8/1024/1024 || 'GB' as shared_buffers;
shared_buffers
-----
32GB
(1 row)

postgres=# \! grep -E "^shared_buffers" /etc/postgresql/15/main/postgresql.conf
shared_buffers = 128MB          # min 128kB
```

Le paramétrage de Postgres (bis)

```
postgres=# select name, source, sourcefile
from pg_settings
where name ~ 'shared_buffers';
   name   |   source   |          sourcefile
-----+-----+-----
shared_buffers | configuration file | /var/lib/postgresql/15/main/postgresql.auto.conf
(1 row)

postgres=# \! cat /var/lib/postgresql/15/main/postgresql.auto.conf
# Do not edit this file manually!
# It will be overwritten by the ALTER SYSTEM command.
shared_buffers = '32GB'

postgres=# alter system reset shared_buffers;
ALTER SYSTEM
```

Cant' replicate!

Quelqu'un que vous ne connaissez pas vous dit:

Je ne sais pas ce qu'il s'est passé. Ma réplication Postgres s'est arrêté. Je jure que je n'ai rien changé. Mon utilisateur pour réplication s'appelle replicator et son mot de passe est r3pl1c4t0r.

Que faites-vous?

Ce que l'utilisateur essaye de faire

```
laetitia:~/03-CantReplicate|⇒ ./spin_up_postgres.sh

[...]

laetitia:~/03-Corrupted|⇒ docker exec -it 03-cantreplicateprincess /bin/sh

$ /usr/lib/postgresql/15/bin/postgres -D /var/lib/postgresql/15/main \
-c config_file=/etc/postgresql/15/main/postgresql.conf &
LOG:  consistent recovery state reached at 0/2000100
LOG:  database system is ready to accept read-only connections
postgresql/.pgpass" has group or world access;
permissions should be u=rw (0600) or less
FATAL:  could not connect to the primary server: connection to server
at "172.17.0.3", port 5432 failed: fe_sendauth: no password supplied
WARNING: password file "/var/lib/postgresql/.pgpass" has group or world
access; permissions should be u=rw (0600) or less
```

Le fichier de password

```
laetitia:~/03-CantReplicate|⇒ docker exec -it \  
03-cantreplicateprincess /bin/sh
```

```
$ ls -l ~/.pgpass
```

```
-rw-rw-rw- 1 postgres postgres 31 Oct 24 22:29  
/var/lib/postgresql/.pgpass
```

```
$ chmod 600 ~/.pgpass
```

Problème résolu!

Ça repart?

```
laetitia:~/03-CantReplicate|⇒ docker exec -it 03-cantreplicateprincess \  
  /bin/sh  
  
$ /usr/lib/postgresql/15/bin/postgres -D /var/lib/postgresql/15/main \  
> -c config_file=/etc/postgresql/15/main/postgresql.conf &  
$ 2022-10-24 23:14:59.484 UTC [212] LOG:  starting PostgreSQL 15.0 (Ubuntu  
15.0-1.pgdg20.04+1) on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1)  
9.4.0, 64-bit  
  
[...]  
  
LOG:  database system is ready to accept read-only connections  
LOG:  invalid record length at 0/3000148: wanted 24, got 0  
LOG:  started streaming WAL from primary at 0/3000000 on timeline 1
```

C'est à vous!

Il faut un backup

```
laetitia:~/03-CantReplicate|⇒ docker exec -it 03-cantreplicatequeen \  
/usr/sbin/ifconfig eth0 | awk '/inet/ { $2; print $2}'  
172.17.0.3
```

```
laetitia:~/03-CantReplicate|⇒ docker exec -it 03-cantreplicateprincess \  
/bin/sh
```

```
$ rm -rf /var/lib/postgresql/15/main/*  
$ pg_basebackup -w -U replicator -X stream -R -c fast \  
-h 172.17.0.3 -D /var/lib/postgresql/15/main
```

Réparé!

On vérifie

```
laetitia:~/03-CantReplicate|⇒ docker exec -it 03-cantreplicateprincess /bin/sh
```

```
$ /usr/lib/postgresql/15/bin/postgres -D /var/lib/postgresql/15/main \  
-c config_file=/etc/postgresql/15/main/postgresql.conf &
```

```
laetitia:~/03-CantReplicate|⇒ docker exec -it 03-cantreplicatequeen psql -c "SELECT * FROM  
pg_stat_replication" -x
```

```
-[ RECORD 1 ]-----  
pid          | 272  
usesysid     | 24576  
username     | replicator  
application_name | 15/main  
client_addr  | 172.17.0.4  
client_hostname |  
client_port  | 49938  
backend_start | 2022-10-24 23:26:57.384432+00  
reply_time   | 2022-10-24 23:27:17.650712+00
```

Réparé!

It's corrupted!

Quelqu'un que vous ne connaissez pas vous dit:

*Postgres ne démarre pas. Je ne sais pas pourquoi. S'il te plaît, sauve-moi la vie, répare-le! Pour info, il devrait y avoir un backup sous
`/var/lib/postgresql/backups/main.`*

Que faites-vous?

Ce que l'utilisateur essaye de faire

```
laetitia:~/04-Corrupted|⇒ ./spin_up_postgres.sh
```

```
[...]
```

```
$ /usr/lib/postgresql/15/bin/postgres \  
-D /var/lib/postgresql/15/main \  
-c config_file=/etc/postgresql/15/main/postgresql.conf &  
PANIC: could not read file "global/pg_control": read 0 of 296
```

Arg...

Résoudre le problème

```
laetitia:~/04-Corrupted|⇒ ./spin_up_postgres.sh
```

```
[...]
```

```
$ cp -R /var/lib/postgresql/backups/main/* /var/lib/postgresql/15/main/
```

```
$ /usr/lib/postgresql/15/bin/postgres \  
-D /var/lib/postgresql/15/main \  
-c config_file=/etc/postgresql/15/main/postgresql.conf &
```

Réparé!

Prévenir les corruptions silencieuses

- Activer le checksum sur toutes les pages
- Vérifier les-dits checksums au moment du backup
- Dumper les bases régulièrement
- Monitorer les fichiers de logs

THANK YOU!