

Comment déplacer une base Postgres avec zéro downtime ?

Naeva Mallet



**Ingénieure
Infrastructure**

leboncoin

Donner à chacun
le pouvoir de vivre mieux
au quotidien.



leboncoin

en quelques mots

1^{ÈRE} MARKETPLACE FRANÇAISE

800K

nouvelles
annonces postées
chaque jour*

plus d'1 internaute sur 2

se rend sur **leboncoin** chaque mois*

X2

du volume
de transactions
en 1 an*

80M

D'annonces en
permanence sur
le site*

28M

visiteurs uniques
se rendent chaque
mois sur **leboncoin***

2^e

site d'e-commerce
le plus visité
en France*

Amazon 6,9M/jour

leboncoin 6,4M/jour

Cdiscount 1,7M/jour

76

univers de
consommation
BtoB BtoC

Sources :

1 Baromètre Fevad du classement des sites commerce en France- 2ème trimestre 2022

2 Médiamétrie Netratings juin 2024

3 Source interne - Biens de consommation

4 Données internes - Janvier 2023

5 Médiamétrie NetRatings - Août 2022



@leboncointech

Our offices



+700 P&T people

+10 P&T guilds

mix of Mission, Autonomy, Advise Authority and Organization

Dev

Architect

Lead Dev

Product Manager

Data Engineer

Product Owner

Data Scientist

Engineering Manager

QA

Cybersecurity

... and many more!

Quelques stacks avec lesquelles on bosse

back



front



devops



mobile



data

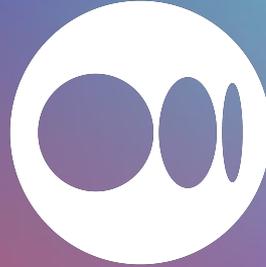


continuous integration





Pour nous
suivre



Medium



Meetup



LinkedIn



Twitch



X (Twitter)

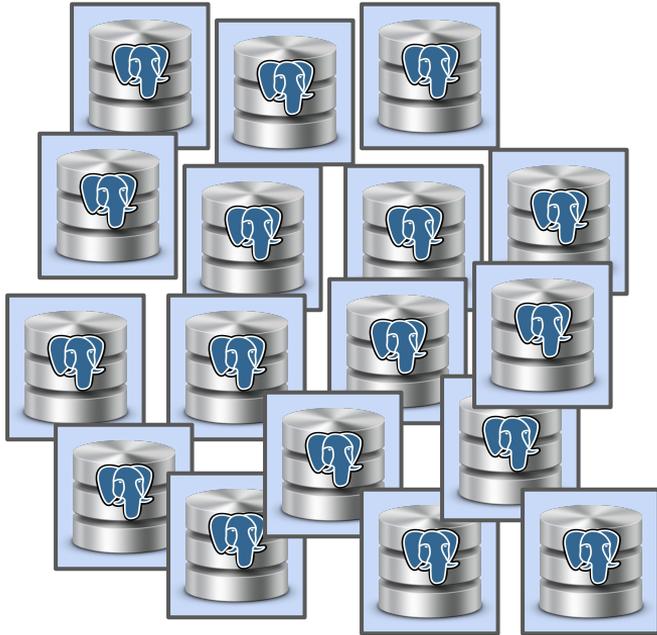


YouTube

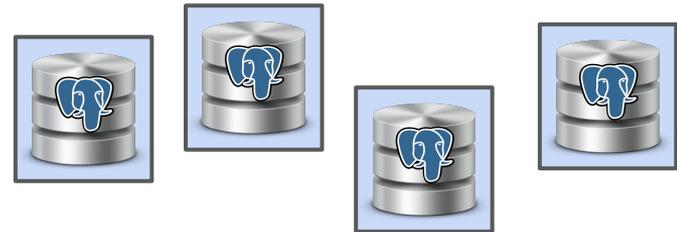
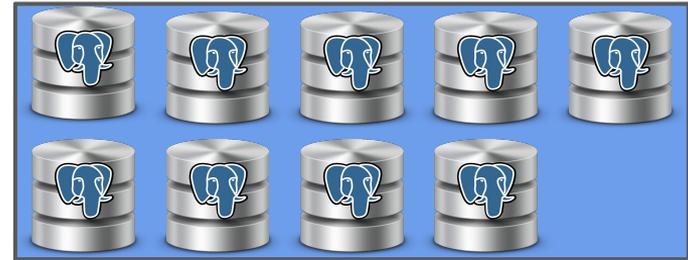
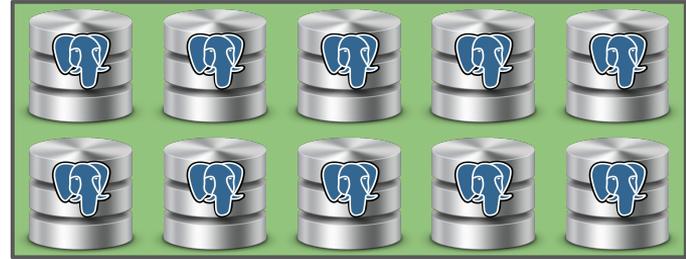
- **Le contexte**
- **C'est quoi la réplication chez PostgreSQL**
- **Ce qu'on a fait**
- **Démo**
- **OpenSource ???**
- **Bilan**



Contexte

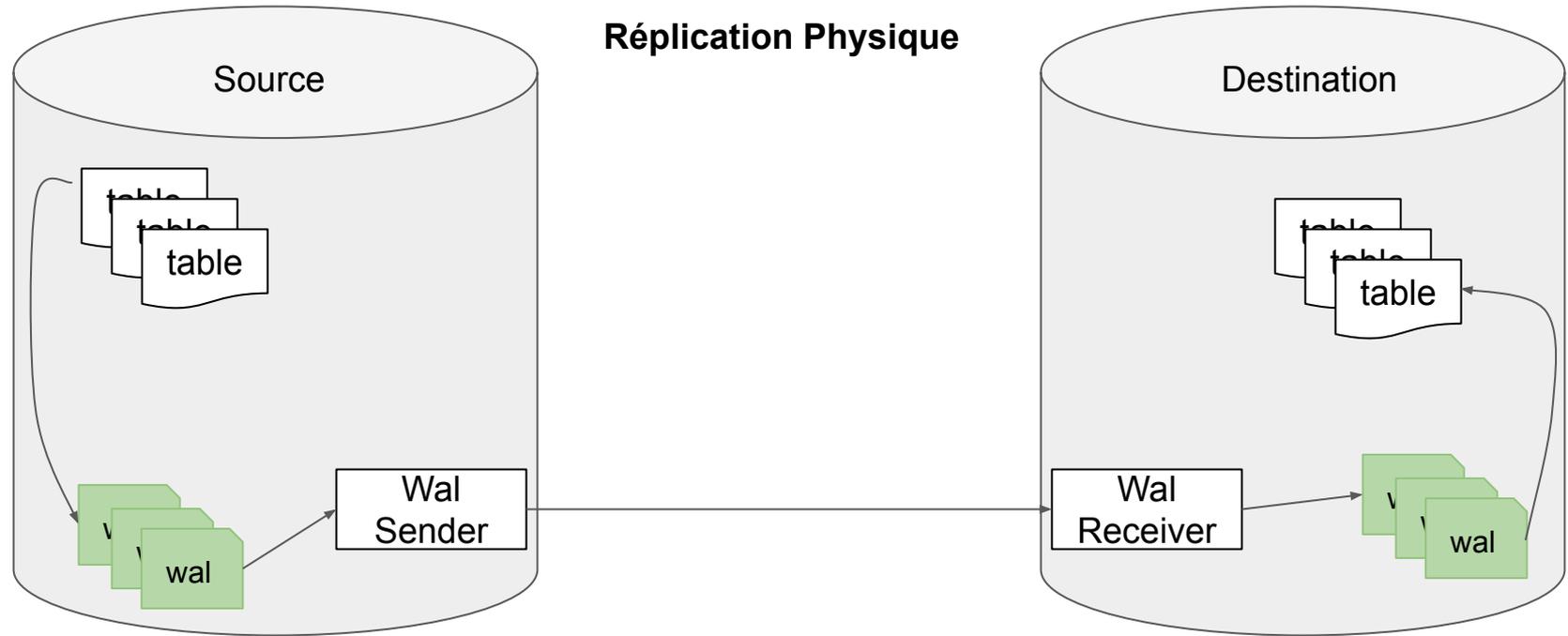


250 bases de données
250 instances dédiées

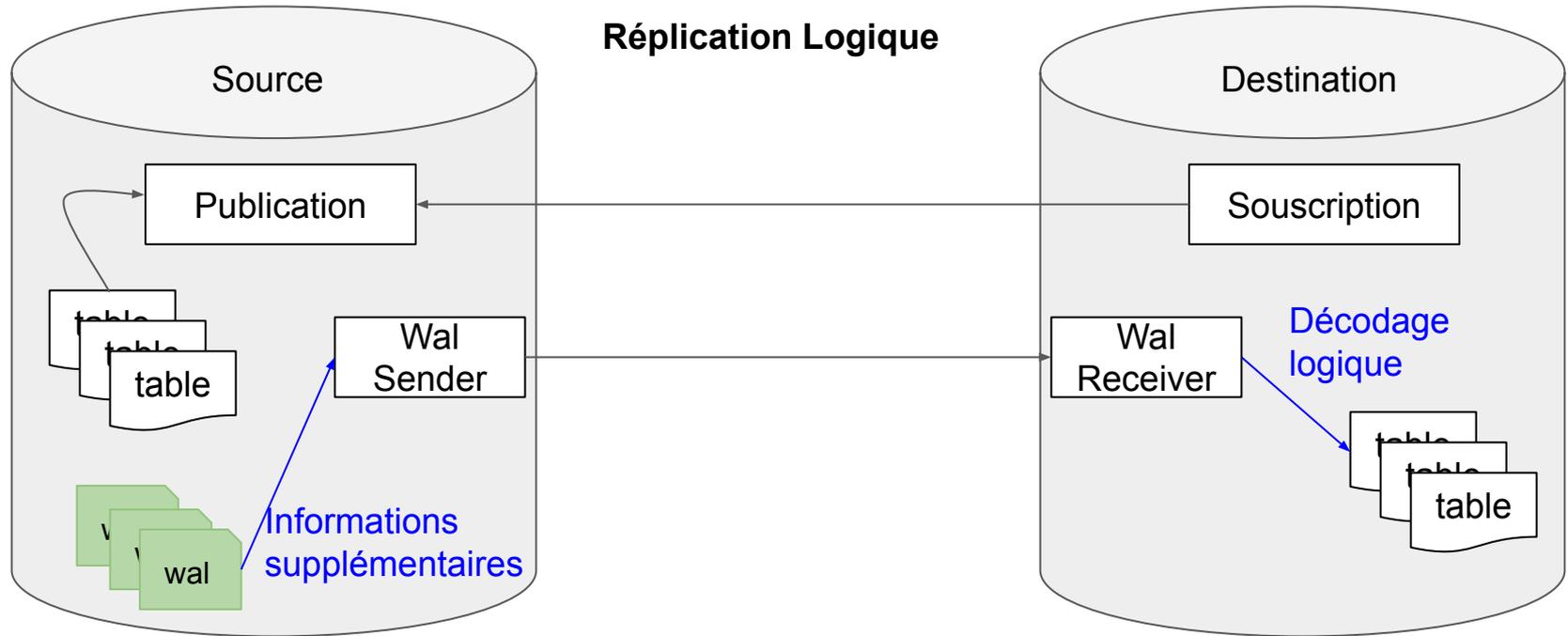


250 bases de données
9 instances mutualisées
96 instances dédiés ou cluster

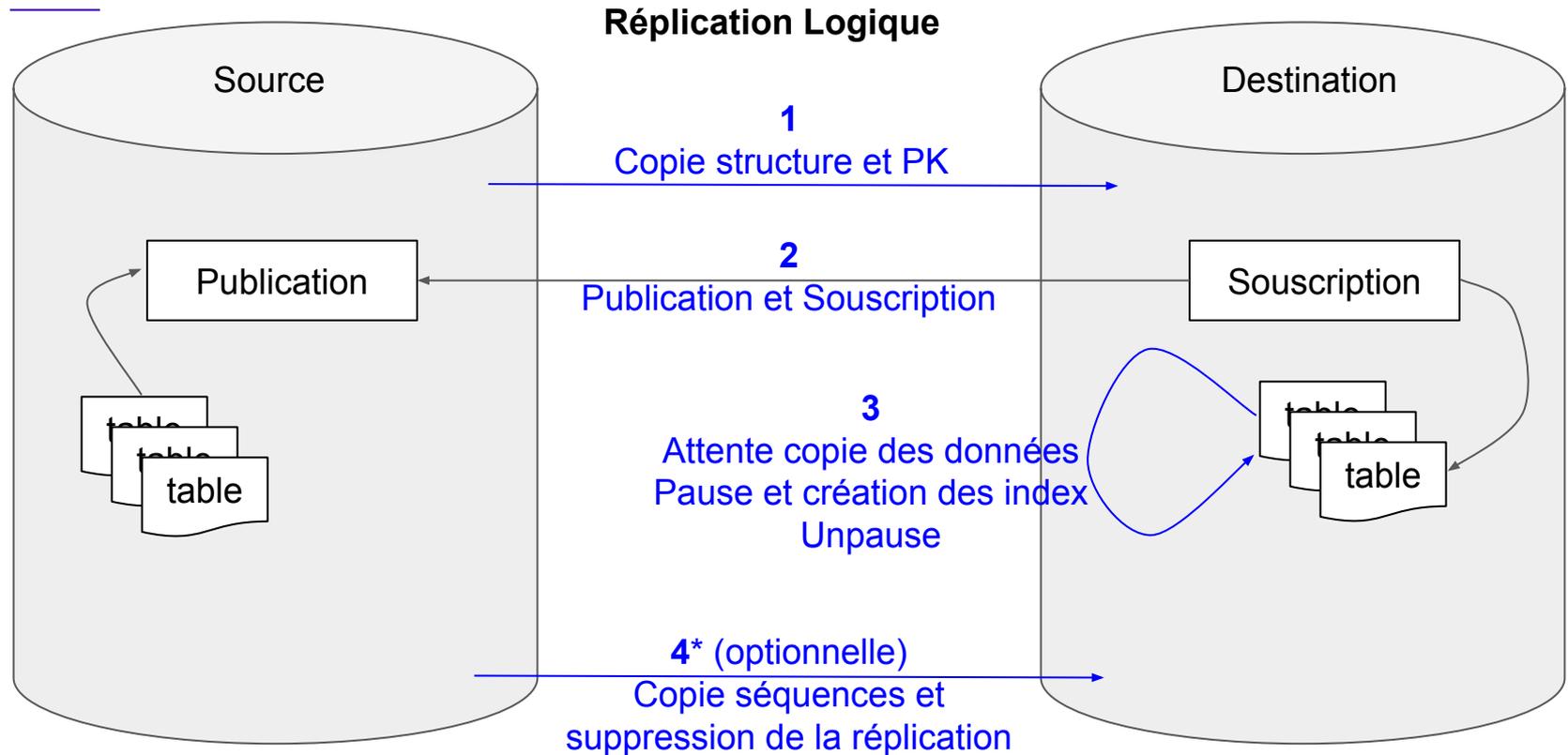
C'est quoi la réplication chez PostgreSQL



C'est quoi la réplication chez PostgreSQL



Ce qu'on a fait



* sera release bientôt

```
1 #!/bin/bash
2
3 SCRIPT=~/.DEV/pg-logical-replication-helper/replication_start.py
4
5 CONN_SRC="postgresql://foo:foopwd@localhost:15431/foo_db"
6 CONN_DST="postgresql://bar:barpwd@localhost:15432/bar_db"
7
8 DB_SRC="foo_db"
9 DB_DST="bar_db"
10
11 FULL_SRC="postgresql://foo:foopwd@pgfoo:5432/foo_db"
12
13 echo "python3 $SCRIPT $CONN_SRC $DB_SRC $CONN_DST $DB_DST"
14
15 # Start the replication process
16 CONN_DB_PRIMARY_FULL=$FULL_SRC python3 $SCRIPT $CONN_SRC $DB_SRC $CONN_DST $DB_DST
```

```
/bin/bash 72x17
$ psql postgresql://foo:foopwd@localhost:15431/foo_db
psql (16.9 (Ubuntu 16.9-1.pgdg20.04+1), serveur 15.13)
Saisissez « help » pour l'aide.
```

```
foo_db=# \dt
```

Liste des relations

Schéma	Nom	Type	Propriétaire
public	table_to_replicate	table	foo
public	table_to_replicate2	table	foo
public	table_to_replicate3	table	foo

```
(3 lignes)
```

```
foo_db=#
```

```
/bin/bash 72x17
$ psql postgresql://bar:barpwd@localhost:15432/bar_db
psql (16.9 (Ubuntu 16.9-1.pgdg20.04+1), serveur 15.13)
Saisissez « help » pour l'aide.
```

```
bar_db=# \dt
```

Aucune relation n'a été trouvée.

```
bar_db=#
```

Starting script : replication_start.py at 20250530-16-52-30

20250530-16-52-30 - Starting pg_dump from server

postgres://foo:foopwd@localhost:15431/foo_db database foo_db 7653 kB

db_schemas : ['public']

db_size : 7653 kB

db_tables : 3

psql "postgres://bar:barpwd@localhost:15432/bar_db" --no-align -tc "select subslotname from pg_subscription where subname like 'subscription_bar_db_%'"

Replication not in progress

20250530-16-52-30 - Starting process : replication_start.py

postgres://foo:foopwd@localhost:15431/foo_db foo_db -

postgres://bar:barpwd@localhost:15432/bar_db database bar_db

create replication user on postgres://foo:foopwd@localhost:15431/foo_db

user replication created

GRANT right on public to replication user

dump section pre-data

pg_dump -d postgres://foo:foopwd@localhost:15431/foo_db -Fp -T public.spatial_ref_sys

--no-acl --no-owner --section=pre-data -n public

pg_restore pre begin

run_dump_restore_pre end

dump section post-data

pg_dump -d postgres://foo:foopwd@localhost:15431/foo_db -Fp -T public.spatial_ref_sys

--no-acl --section=post-data -n public

pg_restore post début

run_dump_restore_post fin

```
Create publication on primary postgresql://foo:foopwd@localhost:15431/foo_db database
foo_db
Add table public.table_to_replicate to publication foo_db_20250530_165230
Add table public.table_to_replicate3 to publication foo_db_20250530_165230
Add table public.table_to_replicate2 to publication foo_db_20250530_165230
Create subscription on secondary postgresql://bar:barpwd@localhost:15432/bar_db database
bar_db
Check if first step of replication is done - db bar_db on host
postgresql://bar:barpwd@localhost:15432/bar_db from
postgresql://foo:foopwd@localhost:15431/foo_db database foo_db
The first step of logical replication is not finished - retrying later
Replication progress : 0/3
Check if first step of replication is done - db bar_db on host
postgresql://bar:barpwd@localhost:15432/bar_db from
postgresql://foo:foopwd@localhost:15431/foo_db database foo_db
Disable subscription on postgresql://bar:barpwd@localhost:15432/bar_db
Restore post section - without primary key
dump section post-data without primary keys
pg_dump -d postgresql://foo:foopwd@localhost:15431/foo_db -Fp -T public.spatial_ref_sys
--no-acl --section=post-data -n public
pg_restore post (without PK) début
pg_restore post (without PK) terminé avec succès
run_dump_restore_post_without_pk fin
Enable subscription on postgresql://bar:barpwd@localhost:15432/bar_db
end=20250530-16-53-42
end
```

```
foo_db=# \x
Affichage étendu activé.
foo_db=# \dRp
Liste des publications
-[ RECORD 1 ]-----+
Nom                | publication_foo_db_20250530_165230
Propriétaire       | foo
Toutes les tables  | f
Insertions         | t
Mises à jour       | t
Suppressions       | t
Tronque            | t
Via la racine      | f

foo_db=#
```

```
bar_db=# \dt
Aucune relation n'a été trouvée.
bar_db=# \x
Affichage étendu activé.
bar_db=# \dRs
Liste des souscriptions
-[ RECORD 1 ]-----+
Nom                | subscription_foo_db_20250530_165230
Propriétaire       | bar
Activé             | t
Publication        | {publication_foo_db_20250530_165230}

bar_db=#
```

Table « public.table_to_replicate »

Colonne	Type	Collationnement	NULL-able	Par défaut	Stockage	Compression	Cible de statistiques	Description
id	integer		not null	generated always as identity	plain			
name	text		not null		extended			
archived	boolean		not null	false	plain			

```
Index :
"table_to_replicate_pkey" PRIMARY KEY, btree (id)
"idx_table_to_replicate_name" btree (name)
Méthode d'accès : heap
```

```
foo_db=# select * from table_to_replicate;
 id | name | archived
----+-----+-----
  1 | test row 1.1 | t
  2 | test row 1.2 | f
(2 lignes)

foo_db=#
```

```
bar_db=# select * from table_to_replicate;
 id | name | archived
----+-----+-----
  1 | test row 1.1 | t
  2 | test row 1.2 | f
(2 lignes)

bar_db=#
```

Bash

Indépendant de
l'infrastructure

Chaîne de connexion

OpenSource

???

Credentials

Traitement des erreurs

Python

<https://github.com/leboncoin/pg-logical-replication-helper>

Bilan

- **Besoin : automatiser la réplication logique**
- **Analyse et solution : Créer un script**
 - **Pour simplifier nos manipulations**
 - **Pour sécuriser nos actions**
- **Résultat : Partager avec la communauté OpenSource**

Questions ?

Merci !

leboncoin
tech