

# CloudNativePG, ou comment embarquer un éléphant sur Kubernetes !

*14 Janvier 2026 - PGSession18  
Pierrick Chovelon*



**kubernetes**



CloudNativePG



PostgreSQL

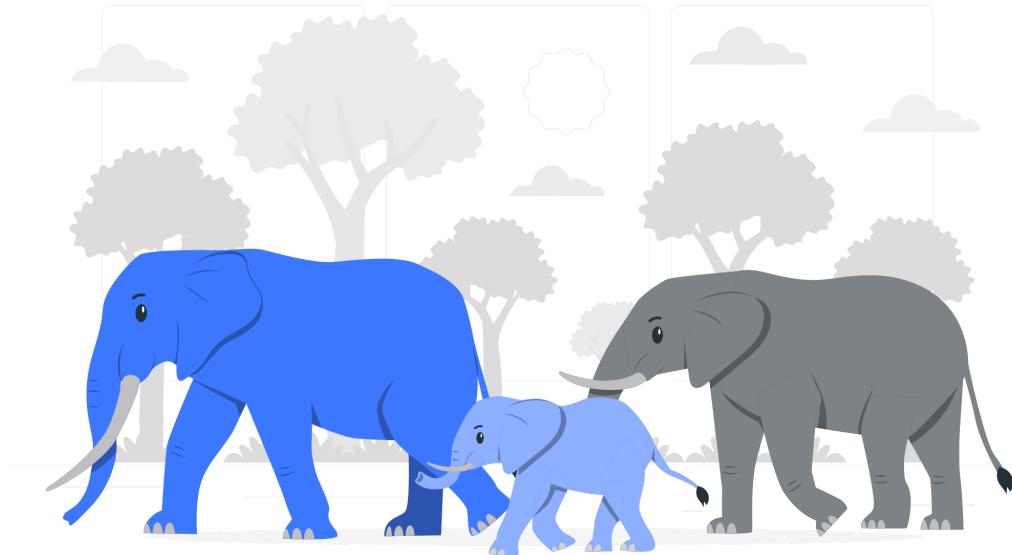
## whoami



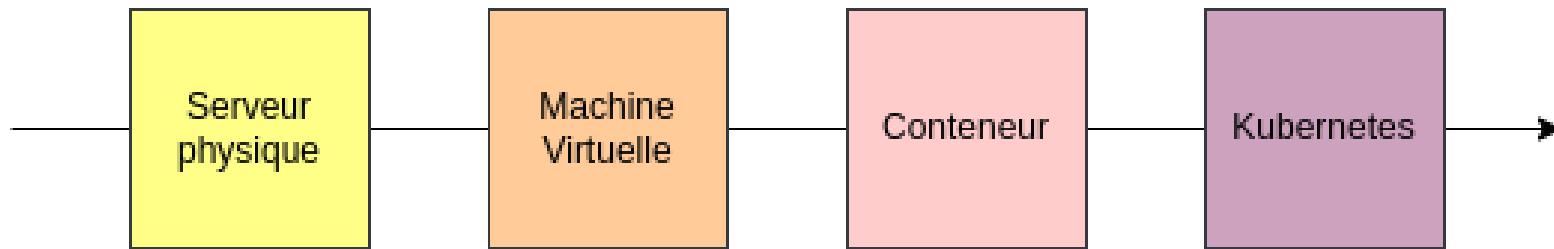
- Pierrick Chovelon
- Consultant Dalibo ~ 3 ans
- Meetup PostgreSQL Lyon
- ... et un DBA embarqué sur Kubernetes !

# Programme

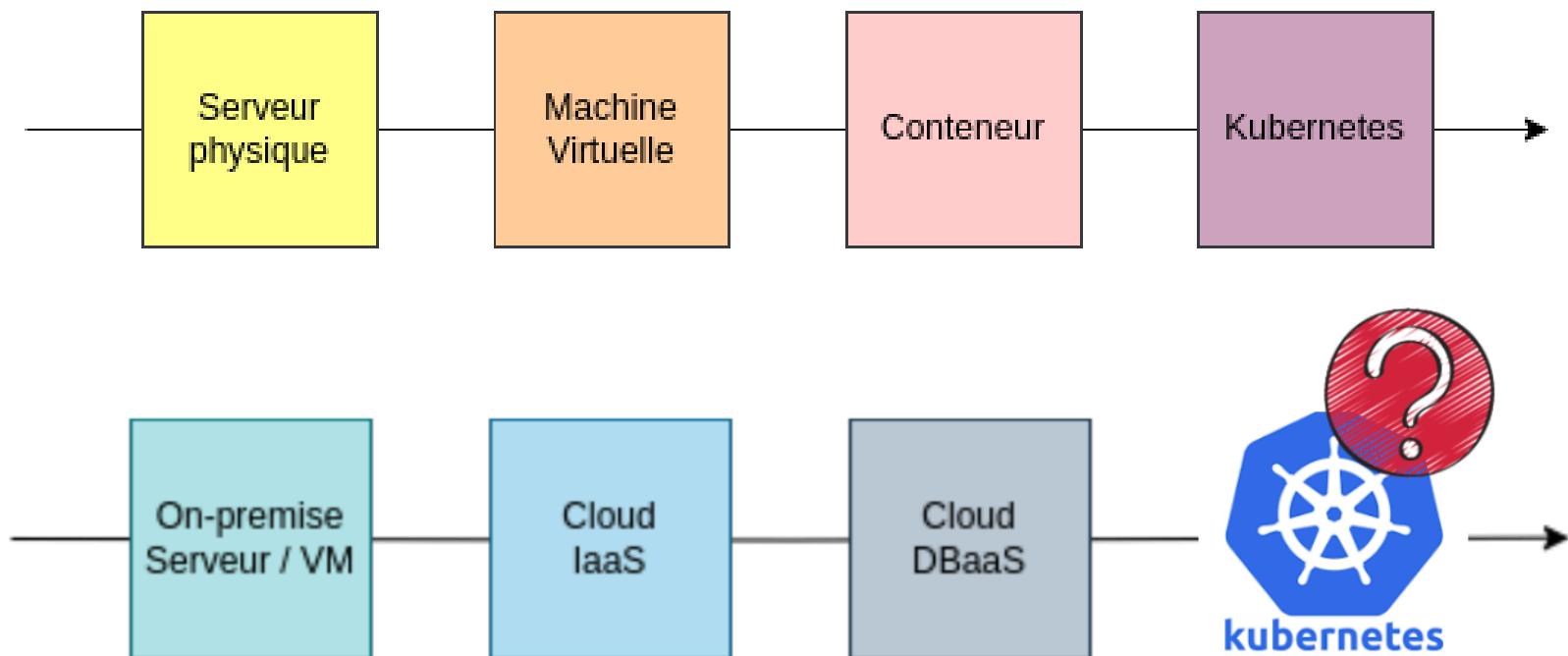
- PostgreSQL dans Kubernetes
- L'opérateur CloudNativePG
- Q&R



# Prenons du recul



# Évolution pour PostgreSQL



# Comment déployer PostgreSQL dans Kubernetes ?

PostgreSQL

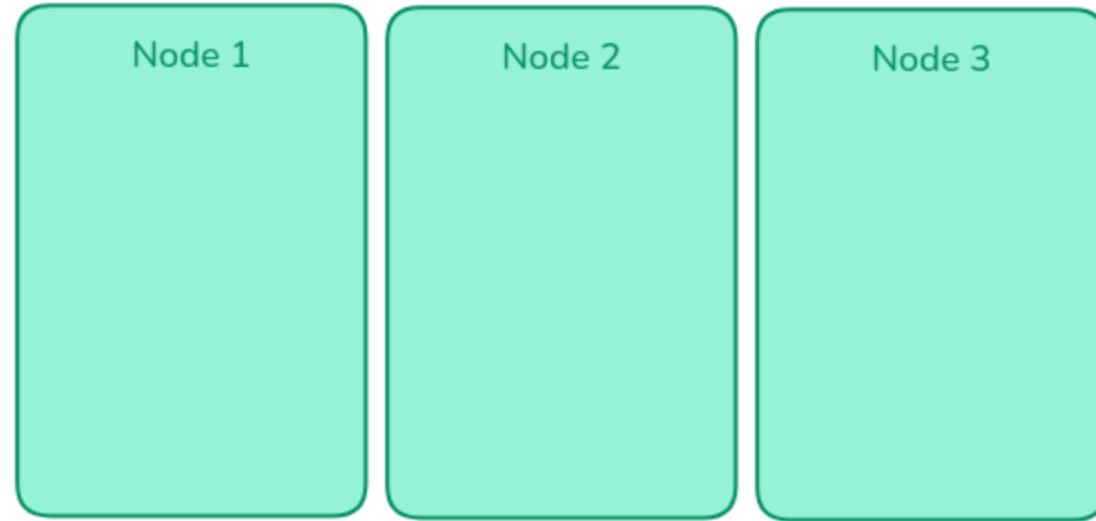
*Binaires PostgreSQL*



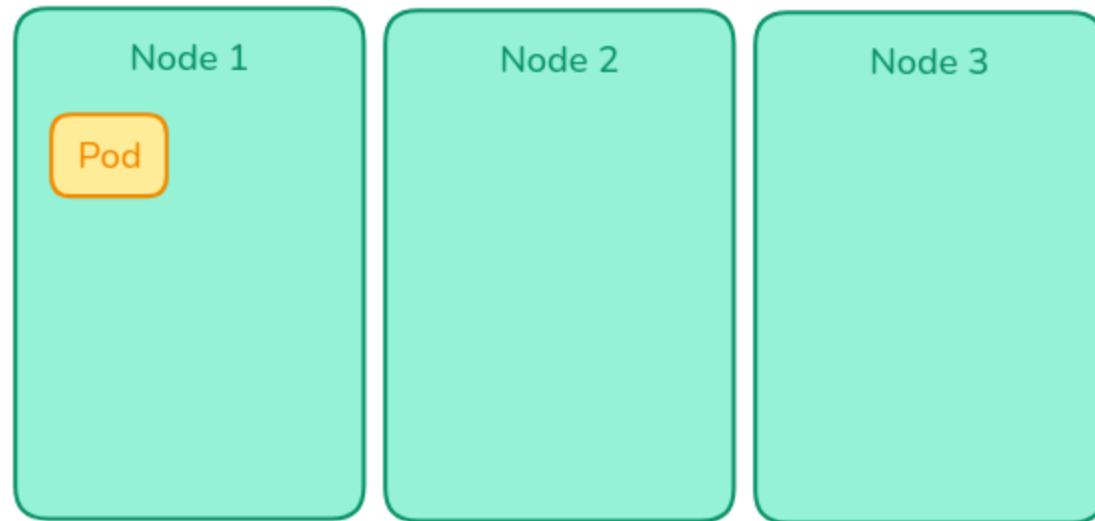
Joli emballage appelé *Image*



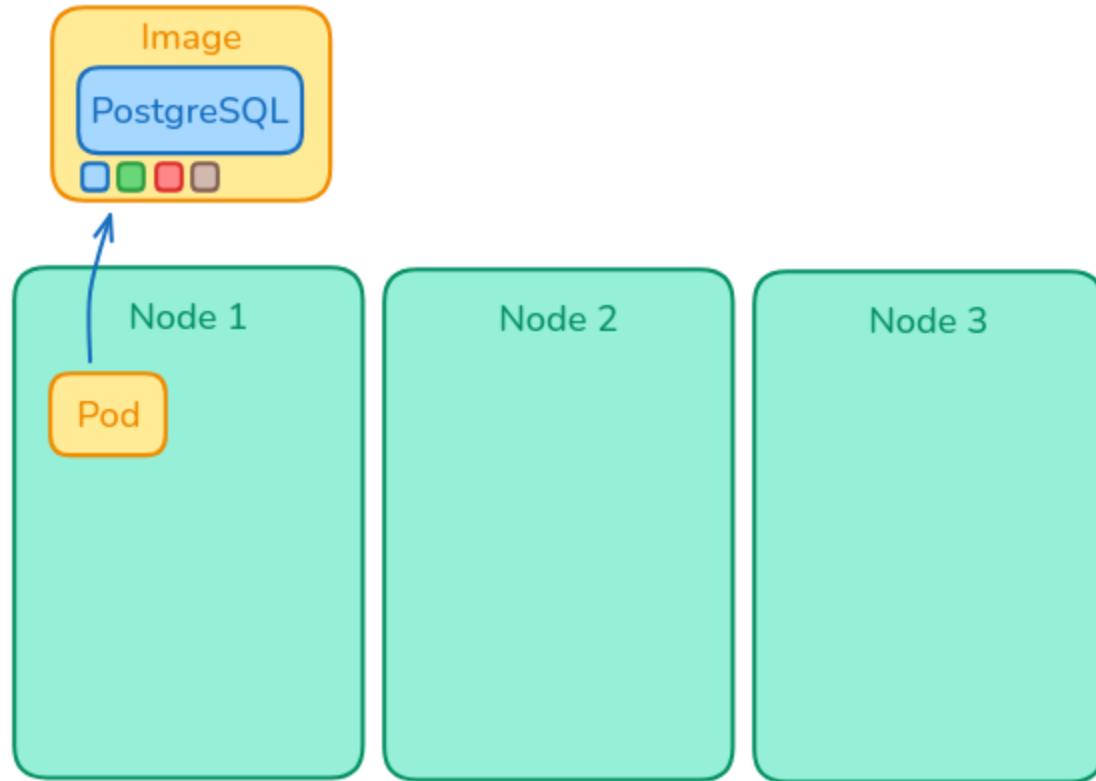
Outils complémentaires



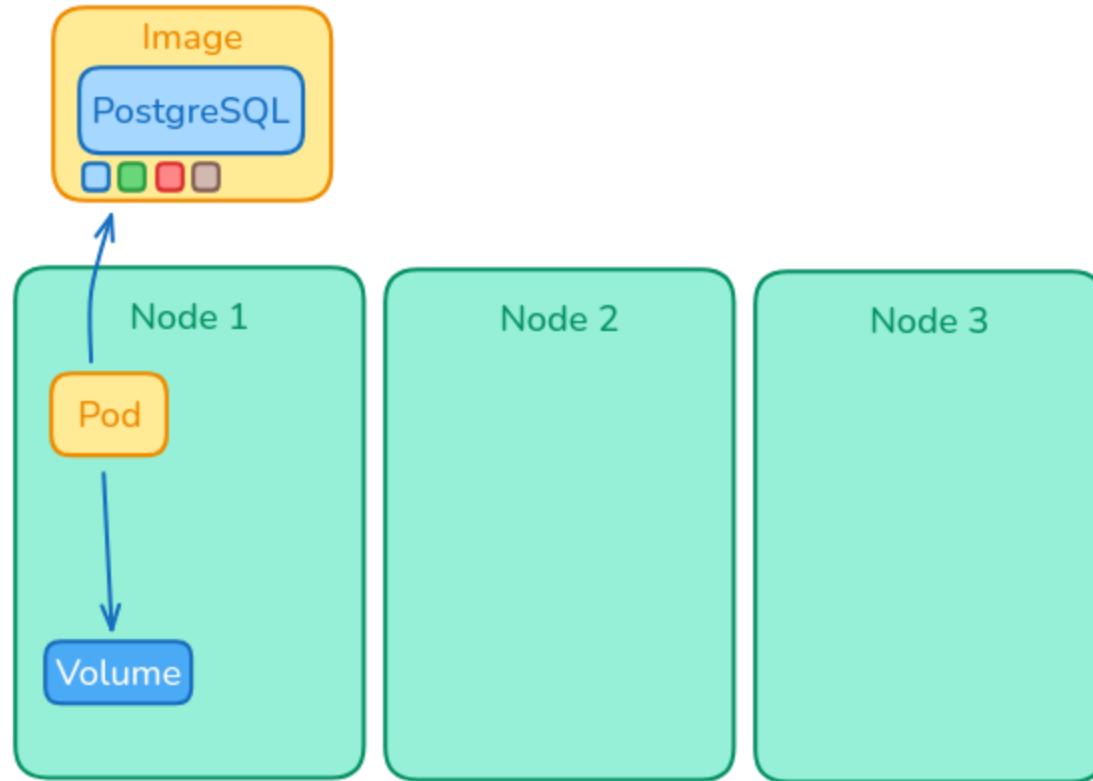
Cluster Kubernetes 3 nœuds



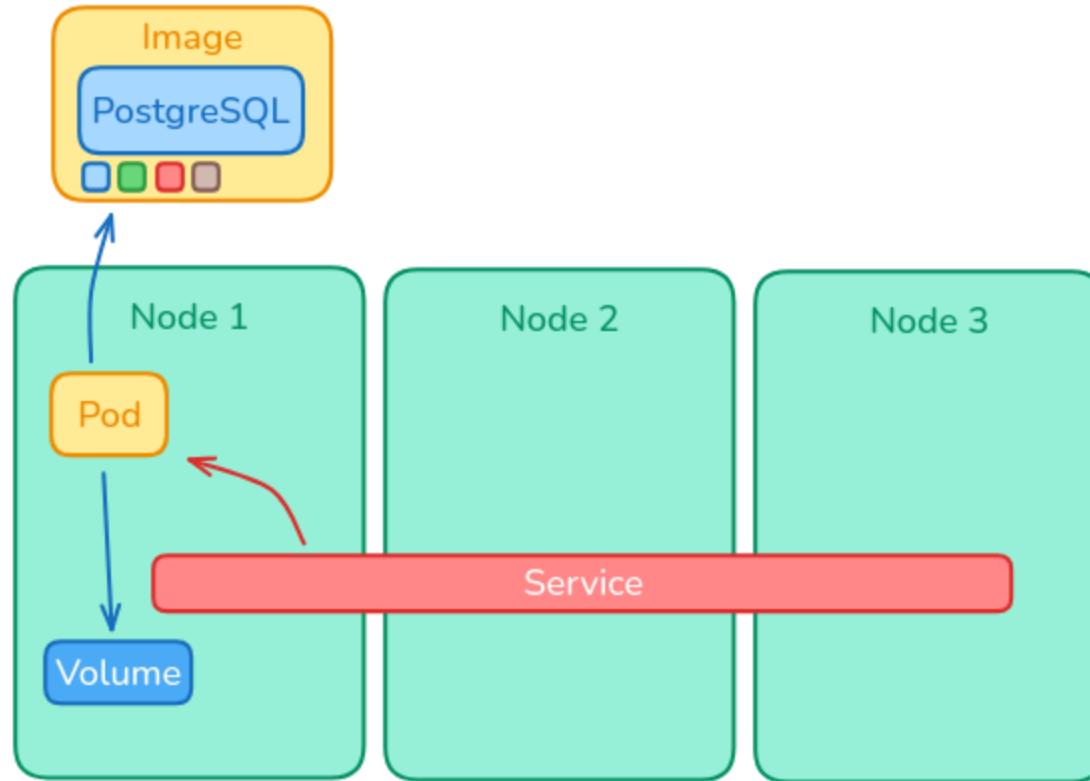
Plus petit élément déployable



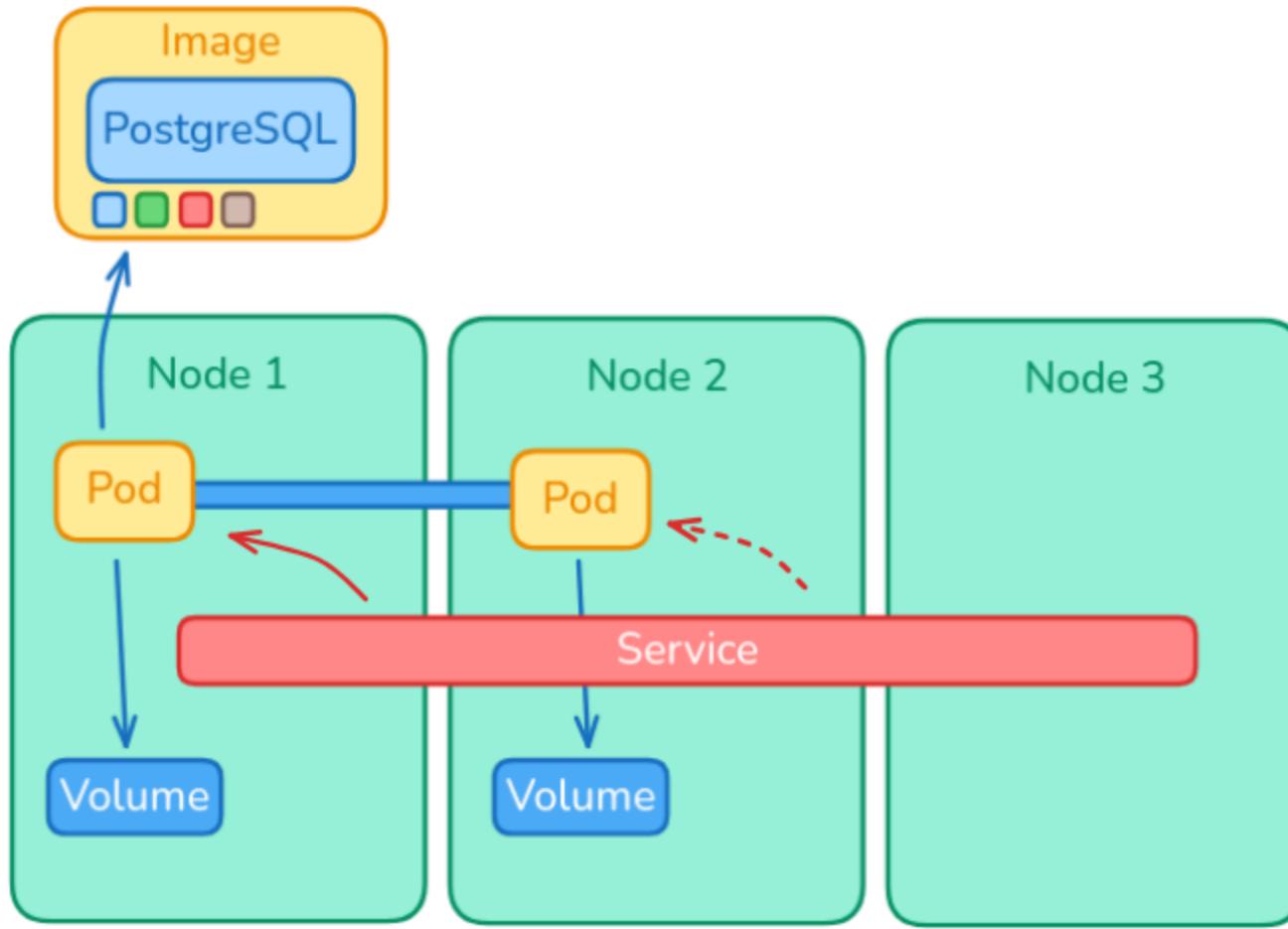
Se repose sur une *Image*



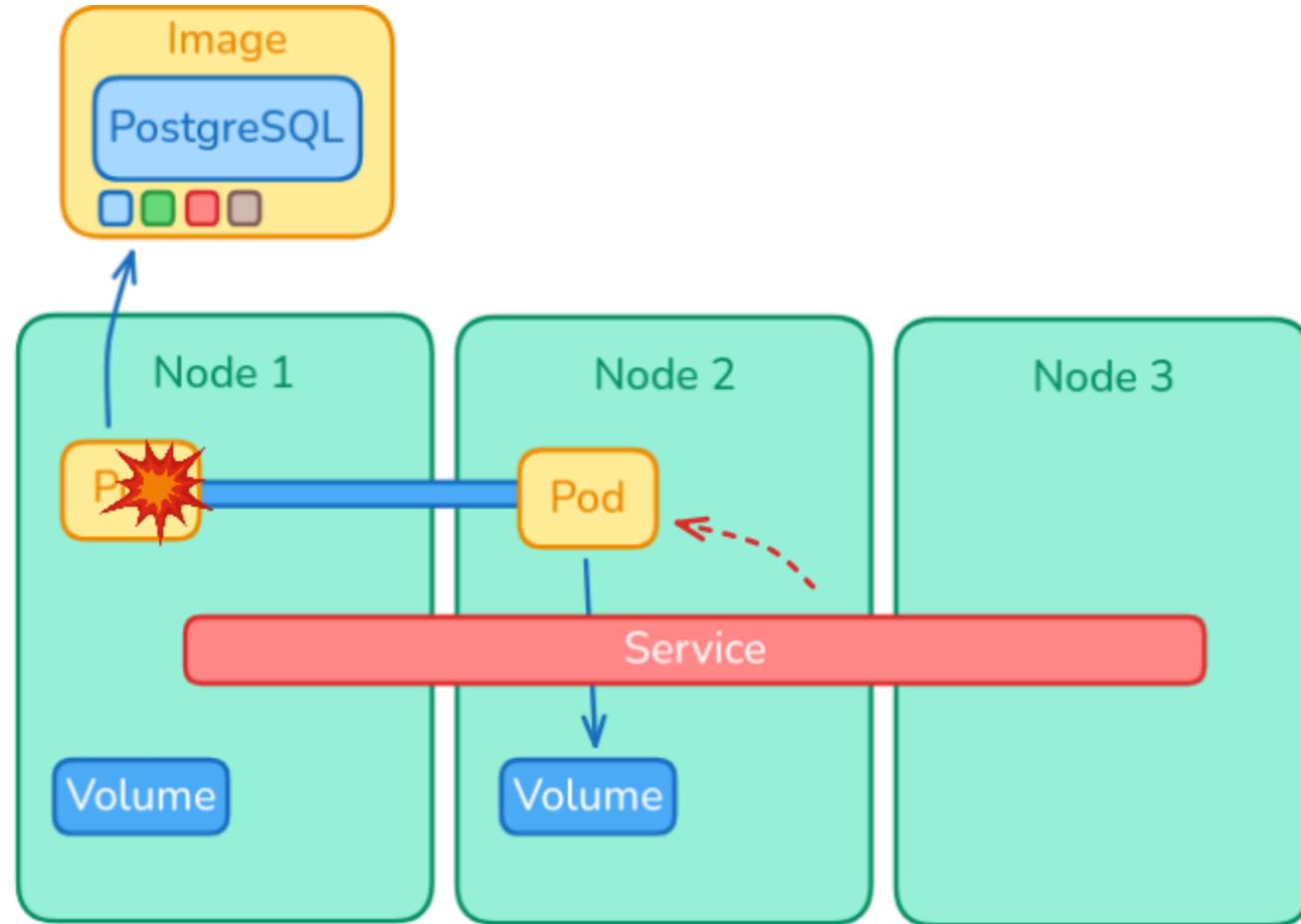
Un peu de stockage pour nos données



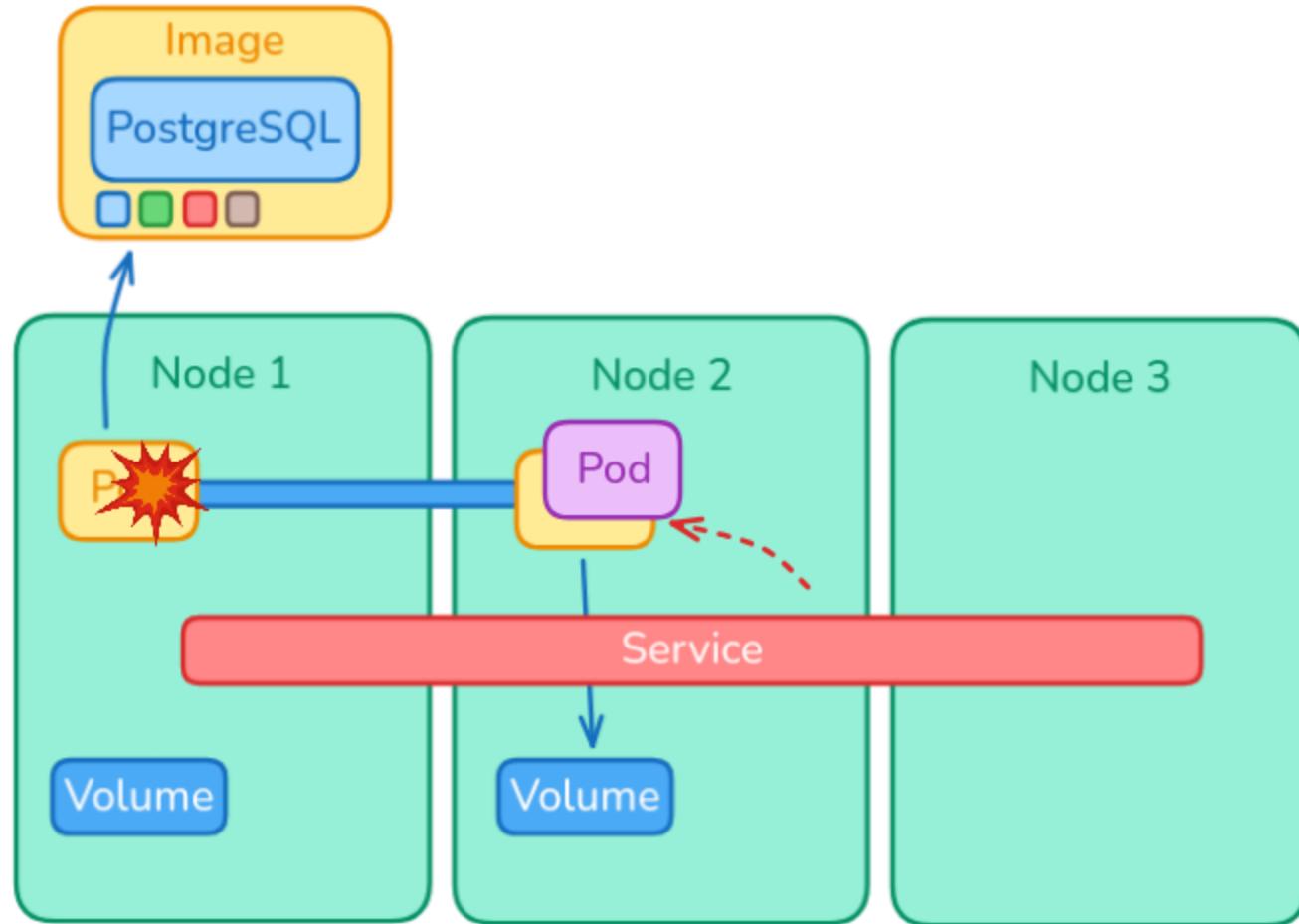
Rendre accessible notre Pod



Un secondaire en réplication 😕



Crash sur l'instance primaire 😱



Quid des montées de version 😱

Et la réponse est ...

Opérateurs Kubernetes 

Et la réponse est ...

Opérateurs Kubernetes 

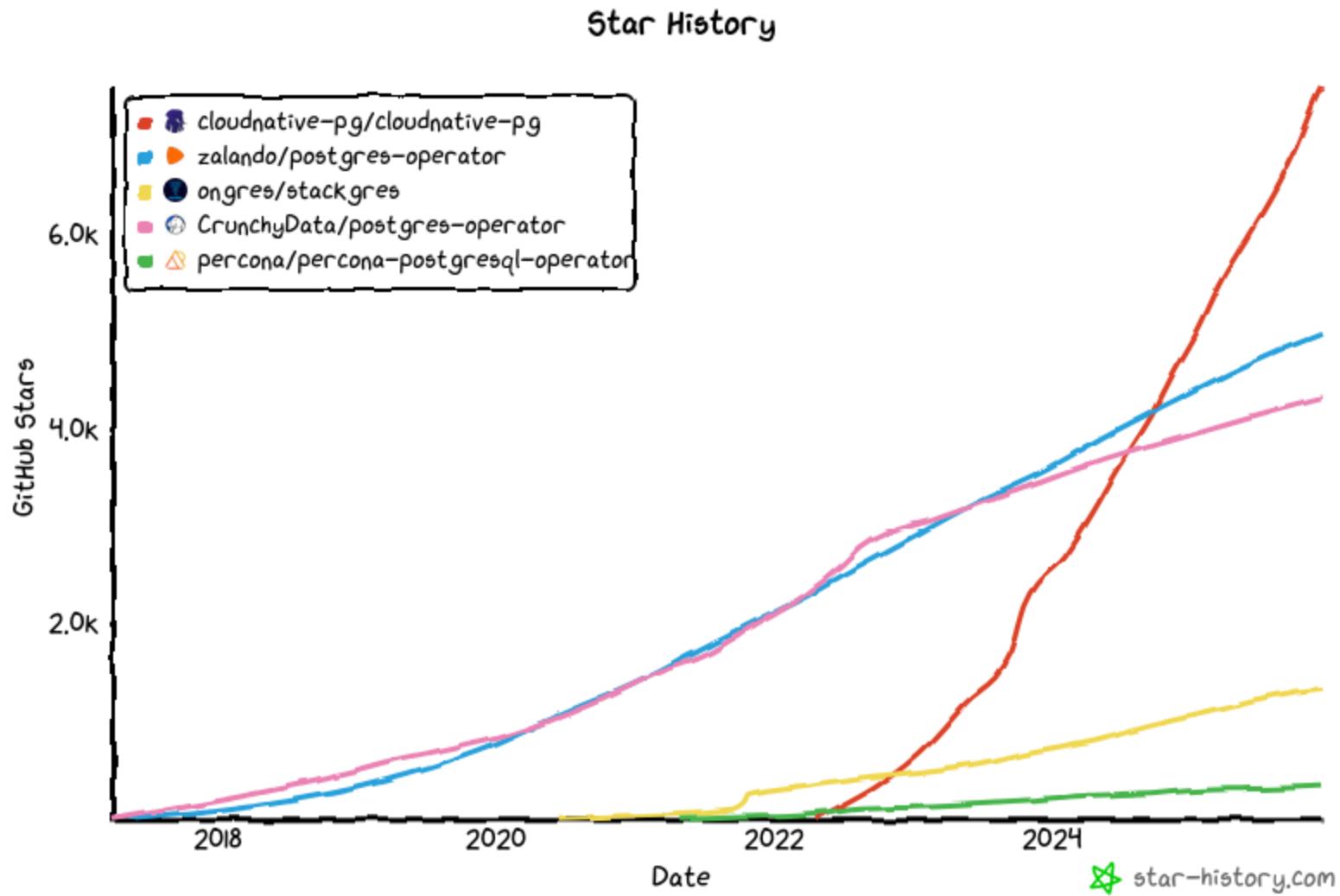


# CloudNativePG

- <https://cloudnative-pg.io>
- Débuté par *2ndQuadrant* puis **EDB**
- Libéré en 2022
- *Open Source*
- Gouvernance similaire au projet PostgreSQL
- Intégré à la Cloud Native Computing Foundation (CNCF)



# Adoption



# Ce que propose CloudNativePG

- Mise à disposition des images
  - Binaires du PGDG
- Déploiement d'instance facilité
  - *Streaming Replication*
- MAJ mineures et majeures automatisées
  - *Rolling Update*
- Sauvegardes physiques PITR
  - Via *plugins*, *Object Storage*, Volume Snapshot
- Bascule (automatique ou manuelle)
- et bien d'autres choses encore ...  

# Principe

- Deux composants
  1. Des nouvelles ressources : extension de l'API Kubernetes (CRD)
  2. Un **controller** , écrit en Go, le cerveau de l'histoire
- Rôles :
  - S'assurer du déploiement des ressources
  - S'assurer du bon fonctionnement de PostgreSQL

# Gestion déclarative des ressources

-  YAML, YAML et encore du YAML
- Décrit ce que l'on veut, laisse faire CloudNativePG
- Cluster , Database , Backup , Publication , ...

# Un cluster mono-instance PostgreSQL

```
apiVersion: postgresql.cnpq.io/v1
kind: Cluster
metadata:
  name: cluster-postgresql
spec:
  imageName: ghcr.io/cloudnative-pg/postgresql:17.6-standard-trixie
  instances: 1
  storage:
    size: 10Gi
```

L'image doit être disponible sur une *registry*.

# Un cluster à 2 instances ( primaire/secondaire)

```
apiVersion: postgresql.cnpq.io/v1
kind: Cluster
metadata:
  name: cluster-postgresql
spec:
  imageName: ghcr.io/cloudnative-pg/postgresql:17.6-standard-trixie
  instances: 2
  storage:
    size: 10Gi
```

primary\_conninfo , slot de réPLICATION, wal\_level

# Configuration PostgreSQL

~~vi postgresql.conf~~ dans le YAML

```
apiVersion: postgresql.cnpq.io/v1
kind: Cluster
metadata:
  name: cluster-postgresql
spec:
  imageName: ghcr.io/cloudnative-pg/postgresql:17.6-standard-trixie
  instances: 2
  storage:
    size: 10Gi
  walStorage:
    size: 5Gi
  parameters:
    shared_buffers: 512MB
    work_mem: 8MB
```

# Une base de données

```
apiVersion: postgresql.cnpq.io/v1
kind: Database
metadata:
  name: db
spec:
  name: db
  owner: app
  ensure: present
  cluster:
    name: cluster-postgresql # on indique le cluster
```

# Métriques

- Exporter Prometheus intégré + métriques customs

```
curl -s http://127.0.0.1:9187/metrics
```

- Dashboard Grafana (ID 20417)



# Traces

```
{  
    "level": "info",  
    "ts": "2026-01-12T13:55:33.679443849Z",  
    "logger": "postgres",  
    "msg": "record",  
    "logging_pod": "cluster-postgresql-1",  
    "record": {  
        "log_time": "2026-01-12 13:55:33.678 UTC",  
        "process_id": "29",  
        "session_id": "6964fd55.1d",  
        "session_line_num": "2",  
        "session_start_time": "2026-01-12 13:55:33 UTC",  
        "transaction_id": "0",  
        "error_severity": "LOG",  
        "sql_state_code": "00000",  
        "message": "starting PostgreSQL 18.1 (Debian 18.1-1.pgdg13+2) ...",  
        "backend_type": "postmaster",  
        "query_id": "0"  
    }  
}
```

Je vous ai menti, il n'y aura pas que du YAML 😠

## On se revoit demain ?

Gestion des versions, montées de version PostgreSQL,  
*Replica Cluster, plugin kubectl, affinité/anti-affinité, ...*

# Conclusion

- Nouvelles opportunités pour PostgreSQL
  - Kubernetes : un **opérateur** est primordial
  - **CloudNativePG**
- **Changements de paradigme**
  - Conteneurs, **méthodes** et **outils** de travail
  - Accompagnement des **DBAs**
- Du temps d'apprentissage
  - Nécessite du temps d'étude et des tests
- Choisir une solution qui répond à vos besoins

## Pour aller plus loin

- Série d'articles de blog  
<https://blog.dalibo.com>
- Site de l'opérateur  
<https://cloudnative-pg.io>
- [contact@dalibo.com](mailto:contact@dalibo.com)
- Slack CloudNativePG



Merci 

Questions ?