

Socle technique PostgreSQL basé sur CloudNativePG

Dalibo

Sommaire

1. Objectif du Socle CloudNativePG pour PostgreSQL
2. Architecture technique
3. Sécurité
4. Standardisation des clusters PostgreSQL
5. Sauvegarde & restauration
6. Observabilité
7. Documentation
8. Support
9. Comment Dalibo vous accompagne

Objectif du Socle CloudNativePG (CNPG) pour PostgreSQL

Objectifs

L'objectif de notre solution est de permettre :

- Un déploiement **rapide et fiable** de clusters PostgreSQL
- Une exploitation **simplifiée** (ops réduites)
- **L'homogénéisation** des pratiques entre environnements
- Une meilleure **maîtrise de la sécurité** et de la conformité
- Intégration
 - type **GitOps**
 - pile de **supervision et d'observabilité**
 - de CNPG, des projets du Dalibo Labs, ainsi que des composants satellites à PostgreSQL (extensions, pg_back, pgbackrest, PoWa etc)
- De contribuer aux projets de cette solution (corrections, nouvelles fonctionnalités, ...)

Périmètre du Socle CNPG

- Architecture (Kubernetes et CNPG)
- Sécurité (chaîne d'approvisionnement et exploitation)
- Standardisation des clusters PostgreSQL (modèles de clusters)
- Sauvegarde et restauration
- Supervision et observabilité
- Documentation à destination des équipes DBA & Ops
- Accompagnement GitOps
- Support des clusters et de la solution Socle

Architecture technique

Kubernetes comme couche d'orchestration

- Gestion du cycle de vie des pods PostgreSQL
- Haute disponibilité
- Gestion des ressources (CPU, mémoire, stockage)
- Isolation par namespaces

CloudNativePG (CNPG)

- Opérateur Kubernetes pour PostgreSQL
- Gestion déclarative des `Cluster` PostgreSQL
- Failover automatique, gestion des réplicas
- Intégration des sauvegardes PITR
- Intégration de la supervision

Stockage : Adaptation aux StorageClasses

- Sélection de la `StorageClass` adaptée selon le profil du cluster
 - stockage rapide (NVMe, SSD) pour les clusters de production
 - stockage standard pour les environnements dev / staging

Principes GitOps

- Tout changement passe par **Git**
- Validation via **CI** (lint, tests, dry-run)
- Déploiement **automatique** (via ArgoCD par exemple)
- Séparation des environnements par branches ou repositories

Sécurité

Deux axes de sécurité

1. Chaîne d'approvisionnement (Supply Chain)

- Traçabilité des composants
- Intégrité des images

2. Exploitation

- Accès et isolation
- Chiffrement des données en transit
- The Cryptic Elephant (TCE)
- Durcissement

Supply Chain : Gestion des images

- Images PostgreSQL et CNPG versionnées explicitement
- Usage d'images signées
- Registre privé et contrôlé
- Scan de vulnérabilités systématique
 - Trivy, Gype, ...

Supply Chain : Signature & intégrité

- Vérification des **signatures** d'images
- Contrôle d'intégrité des artefacts livrés
- Documentation de la provenance des ressources (builds, images)

Supply Chain : SBOM (Software Bill of Materials)

Inventaire exhaustif des composants de chaque version du Socle.

Couvre :

- Dépendances des images PostgreSQL
- Composants Kubernetes utilisés

Supply Chain : Politique de mise à jour

- Cycle de mise à jour régulier (trimestriel)
- Séparation :
 - patches de sécurité
 - upgrades mineurs PostgreSQL
 - upgrades majeurs CNPG / Kubernetes
- Validation en environnement de staging obligatoire

Sécurité à l'exploitation : Accès & isolation

- RBAC Kubernetes : contrôle d'accès aux ressources
- Rôles PostgreSQL **standardisés**
- Séparation des environnements (namespaces, clusters)

Sécurité à l'exploitation : Chiffrement

- Chiffrement TLS :
 - pour les données en transit
 - ou au repos (en fonction du support par la solution de stockage)
- Gestion des certificats
- Chiffrement des sauvegardes
- Point TCE / intégration The Cryptic Elephant (en cours de construction)

Sécurité à l'exploitation : Durcissement

- Configuration PostgreSQL sécurisée par défaut
- Limitation des superusers
- Audit logging activable

Standardisation des clusters PostgreSQL

Types de clusters

Le Socle définit plusieurs modèles standards :

- **Cluster standalone standard** (production non-critique)
- **Cluster HA standard / lecture intensive** (production classique)
- **Cluster dev / staging** (configuration allégée)
- **Cluster multi-zone / résilience renforcée**
- **Cluster selon besoin métier** (AI, graphe, NOSQL, Caching, DWH, etc.)

Paramètres standardisés

- Dimensionnement **CPU / mémoire** par profil
- Politique de redémarrage
- Configuration **WAL** et réplication
- Contraintes d'**affinité**
- Limites de ressources Kubernetes

Sauvegarde & restauration

Stratégie de sauvegarde

- Sauvegardes complètes régulières
 - via **Barman** ou **pgBackRest**
- Sauvegardes différentielles et incrémentielles
- Archivage des journaux (WAL) en continu (**PITR** Point-In-Time Recovery)

Observabilité

Métriques collectées

- Connexions actives
- Latence des requêtes
- Replication lag
- Cache hit ratio
- Consommation CPU / mémoire / I/O

Alerting

Notifications configurées sur :

- Saturation disque
- Retard de réplication
- Échecs des sauvegardes
- Anomalies de connexion
- Saturation du nombre de connexions

Tableaux de bord (Dashboards) Grafana

Dashboards **standardisés** pour :

- Exploitation quotidienne
- Analyse de performance
- Capacity planning

Et dashboards **personnalisés** selon les besoins métiers.

Documentation

Dalibo livre une documentation de référence complète, ajustée à vos besoins et votre contexte :

- Documentation technique d'architecture (DAT)
- Documentation utilisateur (MEX)
- Manuel d'Installation (MI)
- Cahier de recette (REC)
- Documentation sécurité & conformité

Support

Support illimité

- Niveaux 1 à 4 sur l'ensemble de l'outillage
- Couvre : CNPG, PostgreSQL, extensions, sauvegardes
- Équipe d'experts **Dalibo**, basée en France

Comment Dalibo vous accompagne

- Deux phases :
 - Build
 - Run

Phase Build : Déroulement

1. **Étude des besoins** et architectures cibles
2. **Ateliers** (visio et/ou présentiel) :
 - Réplication PostgreSQL
 - Sauvegarde et restauration
 - Standardisation des configurations
 - Supervision et observabilité
 - Procédures d'exploitation

Phase Build : Livrables

- Documentation de référence : **DAT**, **MI**, **MEX**
- Configuration standardisée pour vos environnements
- Image ou procédure de base pour construire vos images
- **Cahier de recette** pour valider la conformité

Phase Run : Évolutions continues

Chaque trimestre, Dalibo met à disposition :

- Nouvelles versions d'images de PostgreSQL
- Mises à jour de la documentation
- Ajustements de configuration
- Support DBA & DevOps pour le déploiement et l'exploitation

Objectifs du Socle CNPG

- Construction d'une **documentation complète**
- Construction des **images** (génériques et dédiées métiers)
- Mise à disposition d'un dépôt (images et documentation) pour un temps limité
- **Accompagnement** à la mise en place et à l'utilisation

Synthèse

Le Socle CloudNativePG, c'est :

- **Mise en production** plus rapide et reproductible
- **Standardisation** des instances PostgreSQL
- **Sécurité** renforcée : supply chain, images, exploitation
- **Observabilité**
- **Coûts d'exploitation** réduits
- **Autonomie** des équipes DBA & Ops
- **Documentation** de référence (DAT, MI, MEX) adaptée à votre environnement
- **Support illimité** architecture, administration, exploitation de la solution et de PostgreSQL

Questions ?