

Continuité des activités et récupération après sinistre

PostgreSQL sur Kubernetes

Utilisation de l'opérateur CNPG



Introduction

Assurer une haute disponibilité et une récupération après sinistre pour les clusters PostgreSQL sur Kubernetes.



Haute Disponibilité

Réplicas de base de données

Une configuration avec un master et plusieurs réplicas garantit qu'en cas de défaillance du master, l'un des réplicas peut être promu.



Haute Disponibilité

Surveillance et alertes

Mettre en place des outils de surveillance tels que Prometheus et Grafana pour surveiller la santé de la base de données.

Configurer des alertes pour des métriques critiques telles que l'utilisation de la mémoire, la latence et le temps de réponse.



Haute Disponibilité

Équilibrage de charge

Utiliser des services tels que PgBouncer pour distribuer les charges de lecture entre les réplicas.



Maintenance planifiée

Mises à jour et correctifs

Mettre régulièrement à jour PostgreSQL et l'opérateur CNPG pour obtenir les dernières fonctionnalités et correctifs de sécurité.

Utiliser des scripts CI/CD pour déployer les mises à jour lors des périodes de maintenance planifiées.



Maintenance planifiée

Sauvegardes régulières

Effectuer des sauvegardes régulières pour éviter la perte de données.

Utiliser des outils tels que `pg_dump` ou des instantanés de volume persistant avec Kubernetes.



Récupération après sinistre



Stratégie de sauvegarde et de restauration

Plan de sauvegarde

Planifier et automatiser les sauvegardes régulières des données.

Utiliser `pg_basebackup` pour les sauvegardes complètes et `pg_dump` pour les sauvegardes logiques, stockées dans des emplacements externes sécurisés (S3, GCS).



Stratégie de sauvegarde et de restauration

Plan de restauration

Documenter et tester régulièrement les procédures de restauration pour garantir une récupération rapide après un sinistre.

Automatiser les scripts de restauration de base de données et effectuer des tests réguliers dans des environnements de pré-production.



Stratégie de réplication géographique

Réplication multi-région

Implémenter des réplicas PostgreSQL dans plusieurs régions géographiques pour garantir la disponibilité en cas de sinistre régional.

Utiliser pglogical pour la réplication logique ou WAL-E pour la réplication asynchrone vers des régions secondaires.



Stratégie de réplication géographique

Basculement automatique et manuel

Mettre en place des processus de basculement automatique pour garantir une disponibilité continue.

Utiliser des solutions telles que Patroni ou Stolon pour la gestion automatique du basculement en cas de défaillance.



Scénarios pratiques

Scénario de défaillance de nœud

Un nœud Kubernetes hébergeant une instance PostgreSQL tombe en panne.

L'opérateur CNPG détecte la panne et promeut un réplica en maître. Les connexions sont automatiquement redirigées vers le nouveau maître.



Scénarios pratiques

Scénario de corruption de données

Des données critiques sont corrompues ou supprimées accidentellement.

Utiliser des sauvegardes régulières pour restaurer les données à un état antérieur à la corruption avec une perte de données minimale.



Scénarios pratiques

Scénario de sinistre régional

Un sinistre (par exemple, incendie, panne de courant) affecte toute une région.

Les répliques géographiques dans une autre région prennent le relais, et le trafic est redirigé vers ces répliques.



Conclusion

La mise en œuvre de stratégies robustes de continuité des activités et de récupération après sinistre garantit la disponibilité et la résilience élevées des clusters PostgreSQL sur Kubernetes.



Questions et réponses

Avez-vous des questions ?

