

Replicate Database

1. Định nghĩa

- Quá trình sao chép và duy trì dữ liệu giữa nhiều database để đảm bảo tính sẵn sàng, dự phòng và cân bằng tải.
- Hỗ trợ đồng bộ hóa dữ liệu theo thời gian thực hoặc định kỳ.

2. Lợi ích

- **High Availability & Fault Tolerance:** Hệ thống vẫn hoạt động khi một database bị lỗi.
- **Load Balancing:** Phân phối truy vấn đọc qua nhiều bản sao, giảm tải cho database chính.
- **Disaster Recovery:** Dễ dàng khôi phục dữ liệu khi gặp sự cố, mất mát hoặc lỗi hệ thống.
- **Data Protection:** Bảo vệ dữ liệu khỏi mất mát, lỗi phần cứng, thiên tai.
- **Scalability:** Dễ mở rộng khi nhu cầu tăng, phục vụ nhiều người dùng hơn.
- **Reduced Downtime:** Bảo trì, nâng cấp mà không gián đoạn dịch vụ.
- **Enhanced Data Access:** Người dùng ở nhiều khu vực truy cập dữ liệu nhanh hơn.

3. Nhược điểm

- **Complexity:** Thiết lập và quản lý phức tạp, nhất là khi có nhiều bản sao.
- **Cost:** Tốn thêm tài nguyên lưu trữ, phần cứng, phần mềm.
- **Inconsistency:** Nguy cơ dữ liệu không đồng nhất giữa các bản sao.
- **Reduced Write Performance:** Ghi dữ liệu có thể chậm hơn do phải đồng bộ nhiều nơi.

4. Key Concepts

- **Primary Database:** Database gốc, nơi ghi dữ liệu chính.
- **Replica:** Bản sao của database chính, lưu ở node khác.
- **Change Data Capture (CDC):** Theo dõi thay đổi dữ liệu để đồng bộ hóa.

5. Các loại Database Replication

- **Synchronous Replication:** Dữ liệu luôn đồng nhất giữa các database, đảm bảo an toàn nhưng có thể chậm.
- **Asynchronous Replication:** Dữ liệu được đồng bộ hóa sau, nhanh hơn nhưng có thể có độ trễ.
- **Snapshot Replication:** Sao chép dữ liệu tại một thời điểm, phù hợp với dữ liệu ít thay đổi.
- **Merge Replication:** Các database cập nhật độc lập rồi hợp nhất lại, phù hợp môi trường cộng tác.
- **Real-time Replication:** Đồng bộ hóa liên tục, dùng cho hệ thống cần dữ liệu mới nhất (banking, social network).

6. Các phương pháp Replication

- **Incremental Replication:**
 - Chỉ truyền dữ liệu thay đổi (insert, update, delete) kể từ lần replicate trước.
 - Tiết kiệm băng thông, tăng hiệu năng, phù hợp hệ thống giao dịch lớn.
- **Full Replacement (Full Refresh):**
 - Ghi đè toàn bộ dữ liệu replica bằng dữ liệu mới nhất từ database chính.
 - Đảm bảo đồng bộ tuyệt đối, nhưng tốn tài nguyên với dữ liệu lớn.
- **Upsert Merge:**
 - Kết hợp insert và update, kiểm tra tồn tại trước khi ghi.
 - Tránh trùng lặp, giảm xung đột dữ liệu.
- **Snapshot Replication:**

- Định kỳ chụp snapshot toàn bộ database và replicate sang bản sao.
- Phù hợp dữ liệu ít thay đổi hoặc cần cập nhật định kỳ.

7. Best Practices

- **Chọn phương pháp phù hợp:**
 - Đánh giá nhu cầu về hiệu năng, độ nhất quán, khả năng chịu lỗi.
 - Synchronous cho độ nhất quán cao, asynchronous cho tốc độ.
- **Giám sát & quản lý:**
 - Theo dõi trạng thái replication, phát hiện lỗi, độ trễ, xung đột.
 - Sử dụng công cụ real-time monitoring, audit log replication.
- **Đảm bảo nhất quán dữ liệu:**
 - Với multi-master, cần conflict resolution, versioning.
 - Kiểm tra và xử lý xung đột để giữ dữ liệu chính xác.
- **Tối ưu hiệu năng:**
 - Nén dữ liệu trước khi truyền, giảm băng thông.
 - Lên lịch replication tránh giờ cao điểm, loại bỏ replicate dư thừa.

8. Công cụ & phần mềm phổ biến

- **Database built-in tools:**
 - MySQL, PostgreSQL, Oracle, SQL Server đều có tính năng replication riêng.
- **Purpose-built tools:**
 - Oracle GoldenGate, Attunity Replicate, HVR.
- **ETL tools:**
 - Informatica, Talend, SSIS.
- **CDC tools:**
 - IBM InfoSphere, Oracle GoldenGate, Attunity Replicate.
- **Data integration tools:**

- Rivery, Talend Open Studio, Informatica PowerCenter.
- **Cloud-based services:**
 - AWS Database Migration Service, AWS Data Pipeline.

Cách Triển khai PostgreSQL HA

A. Chuẩn bị cluster AKS

1. Tạo cluster AKS trên Azure Portal / minikube.
2. Cài đặt kubectl, Helm trên máy cá nhân.

B. Triển khai PostgreSQL HA

1. Thêm Helm repo Bitnami:

```
helm repo add bitnami https://charts.bitnami.com/bitnami
helm repo update
```

2. Cài đặt PostgreSQL HA cluster:

```
helm install hms-postgres bitnami/postgresql-ha \
--set postgresql.replicaCount=2 \
--set postgresql.password=postgres123
```

| Tạo 1 primary và 1 replica (có thể tăng replicaCount).

C. Kiểm tra các pods và services

```
kubectl get pods -l app.kubernetes.io/name=postgresql-ha
kubectl get svc
```

| Đảm bảo các pod và service đều Running.

D. Kết nối ứng dụng với Pgpool-II

Sửa connectionString:

```
Host=hms-postgres-postgresql-ha-pgpool.default.svc.cluster.local;Database=HMS_CentralDB;Username=postgres;Password=postgres123;Port=5432
```

| Pgpool-II sẽ tự động route truy vấn ghi/đọc.

E. Giải thích

1. Bạn đang dùng mô hình gì?

- **Mô hình:** PostgreSQL HA (High Availability) với **asynchronous streaming replication**.

```
SELECT
  client_addr,
  state,
  sync_state,
  sent_lsn,
  write_lsn,
  flush_lsn,
  replay_lsn,
  (pg_current_wal_lsn() - replay_lsn) AS byte_lag
FROM pg_stat_replication;
```

- **Triển khai:** Dùng Helm chart Bitnami `postgresql-ha`.
- **Kiến trúc:**
 - 1 **primary node** (master): nhận ghi (INSERT/UPDATE/DELETE).
 - Nhiều **replica node** (standby): nhận dữ liệu replicate từ primary, chỉ cho phép đọc (SELECT).
 - **Pgpool-II:** Load balancer, tự động chuyển truy vấn ghi đến primary, truy vấn đọc đến replica.

2. Hoạt động ra sao?

- Khi ứng dụng ghi dữ liệu (INSERT/UPDATE/DELETE), Pgpool-II chuyển truy vấn đến primary node.
- Khi ứng dụng đọc dữ liệu (SELECT), Pgpool-II có thể chuyển truy vấn đến replica node (nếu kết nối ở chế độ read-only).
- **Replication:** Primary node sẽ tự động gửi dữ liệu mới sang các replica node (asynchronous, có độ trễ nhỏ).
- Nếu primary node bị lỗi, một replica sẽ được chuyển thành primary mới (failover).

3. Loại database replication bạn dùng là gì?

- **Streaming Replication (Asynchronous):**
 - Primary ghi WAL (Write-Ahead Log), replica đọc WAL và cập nhật dữ liệu.
 - Replica luôn gần như đồng bộ với primary, nhưng có thể trễ vài giây.
 - Phù hợp cho HA, backup, scaling đọc.

4. Giải thích về PostgreSQL HA Helm Chart (Bitnami)

Helm chart này giúp bạn triển khai một **PostgreSQL cluster có tính sẵn sàng cao (HA)** trên Kubernetes, gồm:

- **Primary node** (master): nhận ghi.
- **Replica node(s)** (standby): nhận dữ liệu replicate từ primary, chỉ cho phép đọc.
- **Pgpool-II:** Load balancer, tự động chuyển truy vấn ghi đến primary, truy vấn đọc đến replica.

Tự động cấu hình cluster:

- Khi bạn cài đặt chart, nó sẽ tự động tạo các pod PostgreSQL (primary + replica) và Pgpool-II.

- Chart dùng **StatefulSet** để quản lý các node PostgreSQL, đảm bảo mỗi node có volume riêng.
- Chart tự động cấu hình các tham số replication, user, password, volume, network.

References

- [Viblo: Tăng performance của SQL Database với replication \[Phần 1\]](#)
- [Rivery: Complete Guide to Database Replication](#)
- [Helm: Helm charts](#)
- [ArtifactHub: Bitnami package for PostgreSQL](#)
- [Github: Bitnami](#)
- [Postgresql: High Availability, Load Balancing, and Replication](#)