

# Replica set

22 Tháng Mười 2025 7:20 CH

**Replica set** (tập bản sao) là cách MongoDB chạy **nhiều mongod cùng một cụm** để đảm bảo **high availability (HA)** và **an toàn dữ liệu**.

## Cốt lõi

- **PRIMARY**: node nhận ghi.
- **SECONDARY**: node **sao chép** (replicate) từ PRIMARY qua **oplog** và có thể phục vụ **đọc**.
- **Arbiter (tuỳ chọn)**: chỉ **bỏ phiếu bầu cử**, **không lưu dữ liệu**.

## Nó hoạt động thế nào?

1. Client ghi vào **PRIMARY** → thao tác được ghi vào **oplog**.
2. Các **SECONDARY** “kéo” (pull) oplog về và áp dụng → dữ liệu đồng bộ.
3. PRIMARY lỗi? Các node **bầu cử** (election) tự động chọn PRIMARY mới trong vài giây.

## Lợi ích chính

- **Không gián đoạn**: tự failover khi một node hỏng.
- **Bảo vệ dữ liệu**: có nhiều bản sao.
- **Chia tải đọc**: chuyển một phần truy vấn đọc sang SECONDARY (khi phù hợp).

## Các khái niệm bạn sẽ gặp

- **Write Concern** (ví dụ w: "majority"): mức xác nhận khi ghi.
- **Read Preference** (ví dụ secondaryPreferred): ưu tiên đọc ở đâu.
- **Initial Sync**: lần đồng bộ đầy đủ đầu tiên cho SECONDARY.
- **Priority/Votes**: ảnh hưởng tới **bầu PRIMARY**.

## Thiết lập nhanh (dev)

```
bash Sao chép mã

mongod --replset rs0 --bind_ip localhost --dbpath /data/db
```

```
js

// trong mongosh
rs.initiate()           // khởi tạo cụm 1 node
rs.status()             // kiểm tra trạng thái
```

## Cụm 3 node (tối thiểu cho HA)

```
js

rs.initiate({
  _id: "rs0",
  members: [
    { _id: 0, host: "mongo1:27017" },
    { _id: 1, host: "mongo2:27017" },
    { _id: 2, host: "mongo3:27017" }
  ]
})
```

#### Best practices ngắn gọn

- Dùng **số lượng thành viên bỏ phiếu lẻ** (3, 5, ...).
- Tránh dùng **arbiter** trừ khi thật cần (không có dữ liệu).
- Bật bảo mật (keyFile, user admin).
- Theo dõi bằng `rs.status()` và log.