

rs.initiate()

22 Tháng Mười 2025 7:11 CH

rs.initiate()

rs.initiate() là lệnh trong mongosh để khởi tạo (initialize) một replica set. Nó ghi cấu hình replica set vào local.system.replset, bầu PRIMARY đầu tiên và bật cơ chế nhân bản (replication).

Mục đích

- **High availability:** có PRIMARY–SECONDARY, tự động bầu lại khi node lỗi.
- **Replication:** SECONDARY nhận bản ghi từ oplog của PRIMARY.
- **Đọc mở rộng/analytics:** có thể đọc từ SECONDARY (nếu bật).
- **Tiền đề cho sharding:** mỗi shard là một replica set.

Điều kiện tiên quyết

1. Mỗi mongod phải chạy với tên replica set giống nhau:

- Cờ dòng lệnh: --replSet rs0
- Hoặc mongod.conf:

```
yaml

replication:
  replSetName: rs0
net:
  bindIp: 0.0.0.0
  port: 27017
```

2. Tên host/IP giữa các node truy cập được lẫn nhau (DNS hoặc /etc/hosts).

3. (Khuyến nghị production) Dùng keyFile cho nội bộ cụm và tạo user quản trị.

Cách dùng nhanh

A) Mô trường dev/single-node (đơn giản nhất)

Chạy mongod:

```
mongod --replset rs0 --bind_ip localhost
```

Mở mongosh và khởi tạo:

```
js                                         Sao chép mã

rs.initiate() // tự tạo cấu hình tối thiểu cho localhost:27017
```

Kiểm tra:

```
js                                         Sao chép mã

rs.status()
db.adminCommand({ hello: 1 }) // (thay cho isMaster cũ)
```

B) Cụm 3 node cơ bản (production tối thiểu)

Trên **mỗi máy** chạy mongod với --replSet rs0. Vào **một** node (dự kiến là PRIMARY) chạy:

```
js

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

Sau vài giây, kiểm tra:

```
js

rs.status()
```

Mẹo: số **voting members** là số lẻ (3, 5, ...) để tránh hòa phiếu.

Thao tác thường dùng sau khi init

- Xem config:

```
js

rs.conf()
```

- Thêm member:

```
js

rs.add("mongo4:27017") // thêm SECONDARY
```

- Thêm member ẩn để backup/analytics:

```
rs.add({ host: "mongo4:27017", priority: 0, votes: 0, hidden: true })
```

- Thêm arbiter (chỉ khi cực cần để chẵn thành lẻ, không lưu dữ liệu):

```
js

rs.addArb("arbiter1:27017")
```

- Giáng cấp PRIMARY tạm thời (bầu lại):

```
js
rs.stepDown(60) // nhường PRIMARY ít nhất 60 giây
```

- Cập nhật cấu hình:

```
const cfg = rs.conf();
// ví dụ đổi priority
cfg.members[0].priority = 2;
rs.reconfig(cfg); // chạy trên PRIMARY
```

Lỗi hay gặp & cách xử lý

- replicaSetAlreadyInitialized**: Cụm đã init rồi.

→ Dùng `rs.conf()`/`rs.status()` để xem; muốn sửa thì `rs.reconfig(...)`.
 → Nếu bạn muốn làm lại từ đầu trên một node: dừng mongod, **xóa thư mục dbPath/local/**, chạy lại mongod rồi `rs.initiate()`.

- KeyFile/Authentication mismatch** (SECONDARY không join được):

→ Tất cả node phải dùng **cùng một keyFile** (quyền 600 trên Linux).

- could not find member / name mismatch**:

→ Đảm bảo host trong config (VD: `mongo2:27017`) có thể ping/resolve từ các node khác, và `bindIp` cho phép truy cập.

- Dùng localhost giữa nhiều máy**:

→ Không dùng localhost trong cấu hình nhiều node; hãy dùng tên máy/IP thật.

- not yet initialized**:

→ Chưa chạy `rs.initiate()` mà đã `rs.add(...)`. Hãy init trước.

Best practices tóm tắt

- 3 node voting (hoặc 5) cho HA; tránh arbiter trừ khi bất khả kháng.
- SECONDARY dành cho đọc/backup: đặt `priority:0`, có thể `hidden:true, votes:0`.
- Bật bảo mật: dùng `keyFile`, tạo `user admin`:

```
use("admin")
db.createUser({
  user: "root",
  pwd: "strong_password",
  roles: [ { role: "root", db: "admin" } ]
})
```

- Giám sát: `rs.status()`, log mongod, và metric (Prometheus/Grafana...).