

**Aix-Marseille Université**  
Polytech Marseille - Département Informatique

---

# **Projet CinéExplorer**

Plateforme Web de Découverte de Films

---

## **Livrable 3**

Distribution et Replica Set MongoDB

**Module :** 4A-BDA – Bases de Données Avancées

**Année universitaire :** 2025-2026

**Encadrants :** AL-KHARAZ M. HADDOU BEN DERBAL H.

**Etudiant :** SAHNOUN SALAH EDDINE

# Livrable 3 - Phase 3

## Distribution et Replica Set MongoDB

Projet CinéExplorer - Bases de Données Avancées

## Introduction

Ce document présente les résultats de la Phase 3 du projet CinéExplorer, consacrée à la mise en place d'un Replica Set MongoDB et aux tests de tolérance aux pannes. L'objectif est d'assurer la haute disponibilité des données cinématographiques.

## Table des matières

<b>1 Architecture du Replica Set</b>	<b>3</b>
1.1 Description de l'architecture . . . . .	3
1.1.1 Structure des noeuds . . . . .	3
1.1.2 Caractéristiques techniques . . . . .	3
1.2 Fonctionnement du Replica Set . . . . .	3
1.2.1 Rôles des noeuds . . . . .	3
1.2.2 Avantages de cette architecture . . . . .	3
<b>2 Procédure de Configuration</b>	<b>3</b>
2.1 Étapes de déploiement . . . . .	3
2.1.1 1. Préparation de l'environnement . . . . .	3
2.1.2 2. Démarrage des instances MongoDB . . . . .	3
2.1.3 3. Initialisation du Replica Set . . . . .	4
2.1.4 4. Vérification de la configuration . . . . .	4
2.2 Automation avec scripts . . . . .	4
<b>3 Résultats des Tests de Tolérance aux Pannes</b>	<b>4</b>
3.1 Test 1 : État initial du cluster . . . . .	4
3.1.1 Résultats du test . . . . .	4
3.1.2 Analyse . . . . .	4
3.2 Test 2 : Écriture et vérification de la réplication . . . . .	5
3.2.1 Résultats du test . . . . .	5
3.2.2 Analyse . . . . .	5
3.3 Test 3 : Panne du Primary . . . . .	5
3.3.1 Scénario testé . . . . .	5
3.3.2 Observations . . . . .	5
3.4 Test 4 : Nouveau Primary et temps de failover . . . . .	5
3.4.1 Résultats du test . . . . .	5
3.4.2 Analyse détaillée . . . . .	6
3.4.3 Processus d'élection observé . . . . .	6
3.5 Test 5 : Lecture des données après failover . . . . .	6
3.5.1 Résultats du test . . . . .	6

3.5.2	Analyse . . . . .	6
3.6	Test 6 : Reconnexion et resynchronisation . . . . .	6
3.6.1	Scénario testé . . . . .	7
3.6.2	Observations . . . . .	7
3.7	Test 7 : Double panne (scénario catastrophe) . . . . .	7
3.7.1	Scénario testé . . . . .	7
3.7.2	Observations critiques . . . . .	7
<b>4</b>	<b>Temps de Failover Mesuré</b>	<b>7</b>
4.1	Méthodologie de mesure . . . . .	7
4.2	Résultats des mesures . . . . .	8
4.3	Analyse des performances . . . . .	8
<b>5</b>	<b>Analyse du Comportement</b>	<b>8</b>
5.1	Avec 1 noeud défaillant . . . . .	8
5.1.1	Comportement observé . . . . .	8
5.1.2	Impact sur l'application CinéExplorer . . . . .	8
5.2	Avec 2 noeuds défaillants . . . . .	8
5.2.1	Comportement observé . . . . .	8
5.2.2	Application du théorème CAP . . . . .	9
5.2.3	Impact sur l'application CinéExplorer . . . . .	9
5.3	Tableau comparatif complet . . . . .	9
<b>6</b>	<b>Conclusions</b>	<b>9</b>
6.1	Conclusions techniques . . . . .	9
6.1.1	Réussites démontrées . . . . .	9
6.1.2	Limitations identifiées . . . . .	9
6.2	Bilan final de la Phase 3 . . . . .	9

# 1 Architecture du Replica Set

## 1.1 Description de l'architecture

L'architecture déployée est un **Replica Set MongoDB** composé de 3 nœuds fonctionnant sur la même machine. Cette configuration permet d'assurer la redondance des données et la tolérance aux pannes.

### 1.1.1 Structure des nœuds

- **Nœud 1 (Primary initial)** : Port 27017 - Rôle de coordination des écritures
- **Nœud 2 (Secondary)** : Port 27018 - RéPLICATION synchrone des données
- **Nœud 3 (Secondary)** : Port 27019 - RéPLICATION synchrone des données

### 1.1.2 Caractéristiques techniques

Paramètre	Valeur
Nom du Replica Set	rs0
Nombre total de nœuds	3
Ports utilisés	27017, 27018, 27019
Mécanisme de réPLICATION	Oplog (journal des opérations)
Quorum nécessaire	Majorité (2 nœuds sur 3)
Base de données	imdb_replica

TABLE 1 – Caractéristiques techniques du Replica Set

## 1.2 Fonctionnement du Replica Set

### 1.2.1 Rôles des nœuds

- **Primary** : Seul nœud qui accepte les écritures, réplique les données vers les Secondaries
- **Secondaries** : Reçoivent les données du Primary, servent les lectures (optionnel)
- **Élection** : Processus automatique pour choisir un nouveau Primary en cas de panne

### 1.2.2 Avantages de cette architecture

1. **Disponibilité** : Survie à la panne d'un nœud
2. **Résilience** : Données répliquées sur 3 nœuds
3. **Performance** : Lectures distribuées possibles
4. **Maintenance** : Mise à jour sans interruption de service

# 2 Procédure de Configuration

## 2.1 Étapes de déploiement

### 2.1.1 1. Préparation de l'environnement

```
1 # Creation des repertoires de donnees
2 mkdir -p data/mongo/db-1 data/mongo/db-2 data/mongo/db-3
```

### 2.1.2 2. Démarrage des instances MongoDB

```

1 # Demarrage des 3 instances sur ports distincts
2 mongod --replSet rs0 --port 27017 --dbpath data/mongo/db-1
3 mongod --replSet rs0 --port 27018 --dbpath data/mongo/db-2
4 mongod --replSet rs0 --port 27019 --dbpath data/mongo/db-3

```

### 2.1.3 3. Initialisation du Replica Set

```

1 // Connexion au premier noeud
2 mongosh --port 27017
3
4 // Initialisation du Replica Set
5 rs.initiate({
6   _id: 'rs0',
7   members: [
8     { _id: 0, host: 'localhost:27017' },
9     { _id: 1, host: 'localhost:27018' },
10    { _id: 2, host: 'localhost:27019' }
11  ]
12 })

```

### 2.1.4 4. Vérification de la configuration

```

1 // Vérifier l'état du Replica Set
2 rs.status()
3
4 // Vérifier la configuration
5 rs.conf()

```

## 2.2 Automation avec scripts

Toute la procédure a été automatisée via le script `setup_replica.sh` qui :

1. Nettoie les anciennes instances
2. Crée la structure de répertoires
3. Démarré les 3 instances MongoDB
4. Initialise le Replica Set
5. Vérifie le bon fonctionnement

## 3 Résultats des Tests de Tolérance aux Pannes

### 3.1 Test 1 : État initial du cluster

#### 3.1.1 Résultats du test

```

Capture: test1_etat_initial
Heure: 2026-01-01 20:47:57
=====
Primary: localhost:27017
Secondaires: ['localhost:27018', 'localhost:27019']
=====
```

#### 3.1.2 Analyse

- Primary identifié : localhost :27017

- **Secondaires actifs** : localhost :27018 et localhost :27019
- **État des membres** : Tous en santé (health : 1.0)
- **Conclusion** : Cluster correctement configuré et opérationnel

## 3.2 Test 2 : Écriture et vérification de la réPLICATION

### 3.2.1 Résultats du test

```
Capture: test2_ecriture
Heure: 2026-01-01 20:48:04
=====
Documents insérés: 3
RéPLICATION vérifiée sur 2 port(s)
RéPLICATION complète: OUI
=====
```

### 3.2.2 Analyse

- **Insertion** : 3 documents insérés avec succès sur le Primary
- **RéPLICATION** : Données répliquées sur les 2 Secondaries
- **Vérification** : RéPLICATION complète confirmée
- **Conclusion** : Mécanisme de réPLICATION Oplog fonctionnel

## 3.3 Test 3 : Panne du Primary

```
ss1@ss1-VivoBook-ASUSLaptop-X515EA-F515EA: ~/Documents/800/cineexplorer$ mongosh --port 27017
Current Mongosh Log ID: 6956cb5f6bb70499efde665
Connecting to:          mongosh://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.10
Using MongoDB:          6.0.27
Using Mongosh:          2.5.10
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
-----
The server generated these startup warnings when booting
2026-01-01T20:30:30.614+01:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2026-01-01T20:30:30.758+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2026-01-01T20:30:30.758+01:00: Soft rlimits for open file descriptors too low
-----
test> rs.status()
{
  "ok" : 0,
  "errmsg" : "MongoServerSelectionError: connect ECONNREFUSED 127.0.0.1:27017",
  "code" : 134,
  "codeName" : "MongoNetworkError",
  "details" : {
    "name" : "MongoNetworkError"
  },
  "stack" : [
    "MongoNetworkError: connect ECONNREFUSED 127.0.0.1:27017"
  ],
  "connectionId" : 1
}
test> ■
```

FIGURE 1 – Capture d'écran du test de panne du Primary

### 3.3.1 Scénario testé

Arrêt brutal du processus MongoDB sur le port 27017 (Primary initial) en utilisant Ctrl+C dans le terminal correspondant.

### 3.3.2 Observations

- Arrêt immédiat du service sur le port 27017
- Détection de la panne par les nœuds 27018 et 27019
- Déclenchement du processus d'élection
- Messages "ELECTION" visibles dans les logs

## 3.4 Test 4 : Nouveau Primary et temps de failover

### 3.4.1 Résultats du test

Capture: test4\_nouveau\_primary

Heure: 2026-01-01 20:56:52

TEMPS D'ÉLECTION: 5.0 secondes  
NOUVEAU PRIMARY: localhost:27018  
DONNÉES ACCESSIBLES: OK  
NOMBRE DE DOCUMENTS: 3

### 3.4.2 Analyse détaillée

- **Temps de failover** : 5.0 secondes (excellent)
  - **Nouveau Primary** : localhost :27018 (Secondaire promu)
  - **Données** : Tous les documents préservés (3 documents)
  - **Accès** : Données immédiatement accessibles

### 3.4.3 Processus d'élection observé

1. Détection de l'absence du Primary (timeout heartbeat)
  2. Déclenchement de l'élection parmi les Secondaires
  3. Vote et sélection du nouveau Primary (27018)
  4. Mise à jour de la configuration du Replica Set

### 3.5 Test 5 : Lecture des données après failover

### 3.5.1 Résultats du test

Capture: test5\_lecture  
Heure: 2026-01-01 20:56:54

hears. 2020-01-01 20:00:01

---

---

## LECTURE DEBUTS SECONDARY

---

## LECTURE DEPOSITS & DOCUMENTS ACCESSIONS

### 3.5.2 Analyse

- **Source de lecture** : Ancien Primary (27017) maintenant Secondary
  - **Données accessibles** : 3 documents (intégrité préservée)
  - **Performance** : Lecture réussie depuis un Secondary
  - **Conclusion** : Service de lecture maintenu pendant le failover

### 3.6 Test 6 : Reconnexion et resynchronisation

FIGURE 2 – Capture d'écran du test de reconnexion

### 3.6.1 Scénario testé

Redémarrage du noeud 27017 (ancien Primary) après sa panne.

### 3.6.2 Observations

- Redémarrage réussi du service MongoDB
- Reconnaissance automatique du nouveau Primary (27018)
- Synchronisation des données manquantes via Oplog
- Rejoint le cluster en tant que Secondary
- Temps de resynchronisation : environ 8 secondes

## 3.7 Test 7 : Double panne (scénario catastrophe)

```
=====
SCÉNARIO DOUBLE PANNE:
=====
CÉNARIO DOUBLE PANNE:
* Arrêt simultané de 2 nœuds
* Il n'en reste qu'un nœud actif
* Plus de majorité (2/3) disponible

D'IMPORTANTE ATTENTION:
* Pas d'élection possible
* Cluster en mode lecture seule
* Ecritures impossibles
* Lectures possibles (1 nœud restant actif)

QUESTIONS À OBSERVER:
1. Comment faites pour détecter la perte de quorum?
2. Les lectures fonctionnent-elles?
3. Que se passe-t-il si on tente d'écrire?
4. Comment le cluster réagit-il?

=====
[21:16:05] # Capture sauvegarde: capture_test7_scenario_20260101_211605.txt
# APRÈS AVOIR TESTÉ MANUELLEMENT CE SCÉNARIO,
notez vos observations ci-dessous:

Les observations: Lors du Test 7 (Double Panne), le quorum est perdu car il ne reste qu'un seul nœud actif sur trois (1/3), alors que la majorité requise est de 2. En conséquence : 1) Le nœud survivant (2017) bascule automatiquement de PRIMARY à SECONDARY pour protéger l'intégrité des données. 2) Les écritures sont désormais impossibles (erreur NotWritablePrimary), ce qui prouve que MongoDB priviliege la cohérence sur la disponibilité en cas de partition réseau. 3) Les lectures sont toujours possibles mais le nœud ne peut plus garantir qu'il possède les dernières données du cluster.
Statut: MANUEL
22:00:54] #
```

FIGURE 3 – Capture d'écran du test de double panne

### 3.7.1 Scénario testé

Arrêt simultané de 2 nœuds sur 3 (27018 et 27019).

### 3.7.2 Observations critiques

- Perte du quorum (1 seul nœud actif sur 3)
- Passage automatique en mode lecture seule
- Blocage complet des opérations d'écriture
- Message d'erreur : "NotWritablePrimary"
- Données accessibles en lecture seulement

## 4 Temps de Failover Mesuré

### 4.1 Méthodologie de mesure

Le temps de failover a été mesuré manuellement avec la méthodologie suivante :

1. Démarrage d'un chronomètre au moment de l'arrêt du Primary (Ctrl+C)
2. Surveillance des logs MongoDB en temps réel
3. Arrêt du chronomètre lors de l'apparition du message confirmant le nouveau Primary
4. Répétition du test 3 fois pour validation

## 4.2 Résultats des mesures

Test	Temps mesuré	Observation
Test 3-4 (Panne Primary)	5.0 secondes	Temps optimal
Test 6 (Reconnexion)	8.0 secondes	Resynchronisation
Test 7 (Double panne)	N/A	Pas d'élection possible

TABLE 2 – Résultats des temps de failover mesurés

## 4.3 Analyse des performances

- **Temps de failover** : 5 secondes est un excellent résultat
- **Impact sur l'application** : Interruption minimale des écritures
- **Comparaison aux standards** : Dans la fourchette basse des temps de failover MongoDB
- **Facteurs influençant** : Configuration par défaut, localhost (latence minimale)

## 5 Analyse du Comportement

### 5.1 Avec 1 nœud défaillant

#### 5.1.1 Comportement observé

Aspect	Comportement
Quorum	Maintenu (2 nœuds actifs sur 3)
Élection	Automatique et rapide (5 secondes)
Écritures	Interrompues pendant 5 secondes
Lectures	Continues sans interruption
Données	Intégrité préservée
Récupération	Automatique et transparente

TABLE 3 – Comportement avec 1 nœud défaillant

#### 5.1.2 Impact sur l'application CinéExplorer

- **Utilisateurs** : Presque imperceptible
- **Fonctionnalités** : Brève interruption des écritures seulement
- **Données** : Aucune perte, cohérence garantie
- **Expérience utilisateur** : Très bonne

### 5.2 Avec 2 nœuds défaillants

#### 5.2.1 Comportement observé

Aspect	Comportement
Quorum	Perdu (1 nœud actif seulement)
Élection	Impossible (pas de majorité)
Écritures	Complètement bloquées
Lectures	Possibles en mode dégradé
Données	En lecture seule
Récupération	Intervention manuelle requise

TABLE 4 – Comportement avec 2 nœuds défaillants

### 5.2.2 Application du théorème CAP

- **C (Consistency)** : Prioritaire - écritures bloquées pour préserver la cohérence
- **A (Availability)** : Partielle - seulement les lectures disponibles
- **P (Partition Tolerance)** : Gérée via architecture distribuée

### 5.2.3 Impact sur l'application CinéExplorer

- **Consultation films** : Possible
- **Ajout d'avis** : Impossible
- **Recherches** : Fonctionnelles mais résultats figés
- **Interface** : Doit afficher un message d'avertissement

## 5.3 Tableau comparatif complet

Caractéristique	0 panne	1 nœud HS	2 nœuds HS
État du cluster	Optimal	Dégradé temporaire	Critique
Quorum	3/3 (100%)	2/3 (66%)	1/3 (33%)
Primary	Actif	Changé en 5s	Aucun
Écritures	✓	pause (5s) puis ✓	✗
Lectures	✓	✓	✓ (seulement)
Données	Sécurisées	Sécurisées	Lecture seule
Récupération	-	Automatique	Manuelle requise
Impact utilisateur	Aucun	Minimal	Important

TABLE 5 – Synthèse comparative des différents états

## 6 Conclusions

### 6.1 Conclusions techniques

#### 6.1.1 Réussites démontrées

- ✓ **Configuration réussie** : Replica Set à 3 nœuds opérationnel
- ✓ **Failover rapide** : 5 secondes seulement
- ✓ **Résilience validée** : Survie à la panne d'un nœud
- ✓ **RéPLICATION fonctionnelle** : Données préservées
- ✓ **Transparence** : Peu d'impact sur les applications

#### 6.1.2 Limitations identifiées

- ! **Double panne critique** : Écritures impossibles
- ! **Ressources multipliées** : Données dupliquées 3 fois
- ! **Complexité accrue** : 3 services à monitorer

## 6.2 Bilan final de la Phase 3

La Phase 3 a été complétée avec succès avec les réalisations suivantes :

- **T3.1** : Architecture Replica Set déployée et validée
- **T3.2** : Tests de tolérance aux pannes réalisés et documentés
- **T3.3** : Intégration Django préparée pour la Phase 4

Les résultats démontrent une excellente tolérance aux pannes avec un temps de failover de seulement 5 secondes, garantissant une haute disponibilité pour l'application CinéExplorer.