CHECKLIST

MongoDB Replica Set Failover Checklist

A practical checklist for detecting MongoDB replica set failover, confirming the elected primary, recovering safely, and verifying application readiness before returning to normal service.

MongoDB Replica Set Failover: Detect and Recover Fast

Purpose

Use this checklist during planned maintenance or unexpected replica set failover to quickly confirm election status, validate cluster health, and verify that applications can write and reconnect cleanly.

---

1) Detect the failover event

- [ ] Confirm whether writes are failing, timing out, or retrying in the application layer.
- [ ] Review database logs for election messages, stepdown events, or primary state changes.
- [ ] Check whether any replica set member reports unreachable, recovering, or stale status.
- [ ] Verify whether the event aligns with maintenance, reboot, network change, or certificate update.
- [ ] Correlate database symptoms with infrastructure signals such as host health, network reachability, and storage latency.

If clients report failures but the database still shows a healthy primary and majority, investigate the connection path or client topology handling first.

---

2) Confirm the elected primary

- [ ] Identify the current primary.
- [ ] Confirm the primary is writable.
- [ ] Verify the replica set has majority support.
- [ ] Check whether the old primary is demoted, isolated, or still reachable.
- [ ] Confirm the other members agree on the current topology.
- [ ] Review replication lag and ensure the new primary is sufficiently up to date.
- [ ] Check for clock drift, disk pressure, or resource exhaustion on the elected node.

Do not assume the newest heartbeat or fastest response indicates the correct primary. Confirm writability and majority.

---

3) Decide whether the failover was expected

- [ ] If maintenance was planned, verify the stepdown or reboot completed as intended.
- [ ] If the event was unplanned, determine whether it was caused by host failure, process crash, storage issues, or network partition.
- [ ] Check whether the failure affected one node, a quorum path, or a wider infrastructure segment.
- [ ] Note whether failover occurred once or whether repeated elections are happening.
- [ ] Record the timestamp, impacted nodes, and duration of the write interruption.

---

4) Validate client recovery

- [ ] Confirm application clients discovered the new primary automatically.
- [ ] Verify connection pools refreshed their topology view.
- [ ] Test a successful write through the normal application path.
- [ ] Test a read through the expected route or service endpoint.
- [ ] Check for retryable errors, timeouts, or stale endpoint references in application logs.
- [ ] Confirm TLS, authentication, and service discovery are still functioning correctly.

If only a subset of clients recovered, investigate mixed driver behavior, cached endpoints, or load balancer/service mesh routing.

---

5) Check for hidden instability

- [ ] Watch for rollback indicators or replication inconsistencies.
- [ ] Verify replication lag is decreasing or stable.
- [ ] Confirm there are no repeated elections.
- [ ] Look for uneven connectivity between members.
- [ ] Check for storage latency, CPU saturation, or memory pressure on all members.
- [ ] Validate that the old primary is not attempting to rejoin in an unstable state.

---

6) Recover safely

- [ ] Stabilize the environment before making changes.
- [ ] Avoid restarting multiple nodes at once unless explicitly required.
- [ ] Do not force another election until the cause of the first event is understood.
- [ ] If the failover was planned, confirm catch-up replication before closing maintenance.
- [ ] If the failover was unplanned, retain evidence and preserve logs for root-cause analysis.

- [ ] Return to normal operations only after the cluster and application path both validate successfully.

---

7) Final verification before closing the incident

- [ ] A writable primary is active.
- [ ] The replica set has majority.
- [ ] No member is in a recovering, unreachable, or unstable state.
- [ ] Client writes succeed through the normal path.
- [ ] Client reads behave as expected.
- [ ] No repeated elections or rollback warnings are present.
- [ ] Monitoring alerts have cleared or been acknowledged with a documented explanation.
- [ ] The incident timeline and root cause hypothesis are documented.

Fast decision guide

Treat as a likely connection-path issue if:

- [ ] The database has a healthy primary and majority.
- [ ] Writes fail only from some clients or some network zones.
- [ ] TLS, authentication, or service discovery changed recently.
- [ ] The replica set election completed cleanly.

Treat as an infrastructure or quorum issue if:

- [ ] No writable primary is available.
- [ ] The set cannot maintain majority.
- [ ] Elections repeat or flap.
- [ ] A member is isolated, unreachable, or lagging significantly.

Post-failover notes template

- Event type: Planned / Unplanned
- Start time: _____
- End time: _____
- Impacted members: _____
- New primary: _____
- Client impact observed: _____
- Likely cause: _____
- Validation completed by: _____
- Follow-up actions: _____

Minimum acceptance criteria for returning to service

- [ ] The elected primary is confirmed and writable.
- [ ] Majority is stable.
- [ ] Client writes succeed.
- [ ] Application reconnects without manual intervention.
- [ ] No active instability is visible in logs or metrics.
- [ ] The root cause is understood well enough to prevent immediate recurrence.

Quick reminder

A failover is not fully recovered just because a new primary exists. Recovery is complete only when the cluster is healthy, clients have reconnected, and no hidden instability remains.