

SQL Server Database Mirroring — covering **prerequisites, real-time use cases, advantages, disadvantages, and common issues with resolutions** — all explained in a practical DBA perspective.

1. What is Database Mirroring?

Database Mirroring is a **high-availability and disaster recovery (HADR)** feature in SQL Server that maintains **two copies** of a single database on **two different SQL Server instances** — the **Principal** and the **Mirror**.

👉 Any transaction committed on the **Principal** is immediately sent to and committed on the **Mirror** (synchronously or asynchronously).

2. Prerequisites for Setting Up Database Mirroring

To successfully configure mirroring, certain **infrastructure, configuration, and version requirements** must be met.

◆ SQL Server Version & Edition

- Supported from **SQL Server 2005** to **SQL Server 2016** (deprecated after 2016).
- Works on **Standard Edition (High Safety mode only)** and **Enterprise Edition (High Performance + High Safety modes)**.

◆ Database Requirements

- The database must be in **FULL recovery model**.
- You must have a **recent FULL backup** and **at least one Transaction Log backup** restored on the mirror **with NORECOVERY**.
- Database name on Principal and Mirror **must be identical**.

◆ Network Requirements

- Principal, Mirror, and optionally **Witness Server** must be able to **communicate** over the network (typically port **5022**).
- Use **static IPs or resolvable hostnames**.
- Ensure **firewall ports** are open for mirroring endpoints.

◆ SQL Server Configuration

- Create **database mirroring endpoints** on both servers using:
 - CREATE ENDPOINT Mirroring
 - STATE=STARTED
 - AS TCP (LISTENER_PORT=5022)
 - FOR DATABASE_MIRRORING (ROLE=ALL);
- Endpoints must be accessible by **domain accounts or certificates** (for cross-domain).

◆ Authentication

- Use either:
 - **Windows Authentication** (preferred in same domain), or
 - **Certificate-based Authentication** (for cross-domain or workgroup setups).

◆ Witness Server (Optional)

- Required only for **Automatic Failover** (High Safety with Automatic Failover mode).
- Witness doesn't store data; it helps detect failures and initiate failover automatically.

3. Real-Time Use Cases of Database Mirroring

Here's where DBAs like you actually use mirroring in real environments 

Use Case	Description
1. High Availability for OLTP Systems	Used in small/medium businesses where AlwaysOn Availability Groups are not affordable. Ensures quick failover in critical transactional databases.
2. Disaster Recovery between Data Centers	Asynchronous mode used to mirror databases across geographically separated data centers.
3. Reporting or Backup Offloading (via Snapshots)	You can create Database Snapshots on the mirror to allow read-only reporting access.
4. Interim HA solution during migration	Used as a temporary DR/HA solution before upgrading to AlwaysOn Availability Groups.
5. Application-Level Failover (via Connection String)	Applications can automatically connect to the new Principal using "Failover Partner" keyword.

Example:

Data Source=SQLPrincipal;Failover Partner=SQLMirror;Initial Catalog=MyDB;Integrated Security=True;

4. Database Mirroring Operating Modes

Mode	Type	Description	Supported in
High Safety Mode (Synchronous)	Manual or Automatic Failover	Transactions are committed on both Principal and Mirror before completion. Zero data loss.	Standard & Enterprise
High Performance Mode (Asynchronous)	Manual Failover Only	Principal does not wait for acknowledgment from Mirror. Some data loss possible.	Enterprise Only

5. Advantages of Database Mirroring

Advantage	Explanation
High Availability	Provides quick failover (within seconds).
Automatic Failover Support	With a Witness server, automatic failover is possible.
Zero Data Loss (Synchronous Mode)	Guarantees no data loss in synchronous mode.
Simple to Configure	Easier to set up than AlwaysOn Availability Groups.
Automatic Page Repair	Corrupted pages are automatically repaired from the mirror copy.
Integration with Applications	Transparent failover supported via connection string.

6. Disadvantages of Database Mirroring

Disadvantage	Explanation
Deprecated Feature	Deprecated since SQL Server 2016; replaced by AlwaysOn AG .
Database-Level Only	Works per database, not at instance level — no support for cross-database transactions.
No Read Access on Mirror	Mirror database is not readable (except via snapshots).
Performance Overhead	In synchronous mode, Principal waits for Mirror acknowledgment → increased latency.
Requires Two Instances	Additional licensing and hardware costs.
Manual Failover in Async Mode	High Performance mode supports only manual failover.

7. Common Issues and Resolutions

Issue	Cause	Resolution
Error: Database Mirroring Login Failed	Endpoint not accessible or incorrect authentication.	Verify endpoint ports, firewall, and user permissions.
Mirroring suspended	Network interruption or transaction backlog.	Resume mirroring using ALTER DATABASE <db> SET PARTNER RESUME;
High Transaction Log Growth	Mirror not catching up; logs can't be truncated.	Fix mirror connectivity or reinitialize mirroring.
Principal in Disconnected State	Network or endpoint down.	Restart endpoints or reestablish partner.
Automatic Failover didn't happen	Witness down or quorum lost.	Ensure all three servers (Principal, Mirror, Witness) are online and in sync.
Performance degradation	Synchronous mirroring on high-transaction DB.	Switch to asynchronous mode or increase network bandwidth.
Error 1474 / 1418	Communication failure between endpoints.	Check endpoint URLs, authentication method, and permissions.

8. Monitoring and Maintenance Tips

Task	Command / Tool
Check mirroring status	SELECT mirroring_state_desc, mirroring_role_desc FROM sys.database_mirroring;
Monitor send/redo queues	sys.dm_db_mirroring_connections, sys.dm_db_mirroring_states
Resume suspended session	ALTER DATABASE dbname SET PARTNER RESUME;
Remove mirroring	ALTER DATABASE dbname SET PARTNER OFF;
Automatic page repair	Enabled by default in Enterprise edition.

9. When to Use Mirroring (in 2025 Context)

Although **Database Mirroring** is **deprecated**, it's still useful in:

- **Legacy systems** (SQL Server 2012–2016)
- **Simple DR setups** where AlwaysOn AG or FCI is not viable
- **Low-cost HA environments**

For modern systems (SQL Server 2016+), **AlwaysOn Availability Groups** or **Failover Cluster Instances (FCI)** are the recommended replacements.