

High Availability & Read-Replica Implementation

Introduction

The client required a resilient SQL Server architecture capable of improving database availability, isolating read-only reporting workloads, and providing a scalable foundation for future growth. The existing environment hosted approximately 70 production databases, ranging from 100 MB to 2.5 TB, running on Microsoft SQL Server Enterprise Edition deployed across AWS EC2 instances.

To meet these requirements, an enterprise-grade SQL Server Always On Availability Group (AG) solution was designed and implemented on a Windows Server Failover Cluster (WSFC) using SQL Server Enterprise features. The work followed a phased approach that validated the complete implementation in a dedicated development environment before production rollout, materially reducing deployment risk while exercising every operational scenario in advance.

The delivered solution established a dedicated read-only replica, reduced reporting load on the primary server, improved operational resilience, and produced a repeatable deployment pattern for future high-availability initiatives.

Client Details

Name: Confidential | **Industry:** Finserv | **Location:** USA

Technologies

Database: Microsoft SQL Server Enterprise

High Availability: SQL Server Always On Availability Groups

Clustering: Windows Server Failover Cluster (WSFC)

Cloud Platform: AWS EC2

Storage: Amazon EBS (GP3)

Backup: SQL Server Native Backup & Restore

Replication: Asynchronous Commit

Operating System: Windows Server

High Availability & Read-Replica Implementation

Project Description:

The client is a leading financial services organization operating a business-critical SQL Server estate that supports multiple enterprise applications. As database usage grew, the client's environment surfaced a set of operational and architectural constraints. A single SQL Server instance served both, transactional processing and reporting; concentrating load on one server while offering limited resilience against infrastructure failure. The estate also spanned roughly seventy databases of widely varying size, including several multi-terabyte databases. So introducing replication demanded careful sequencing to minimize production impact and avoid extended maintenance windows.

The engagement was framed around a focused set of objectives:

- Implement SQL Server Always On Availability Groups.
- Introduce a dedicated read-only replica.
- Improve database availability and failover readiness.
- Reduce reporting load on the production server.
- Minimize production downtime during implementation.
- Establish a scalable architecture for future expansion.

Implementation

A phased implementation strategy was selected to minimize deployment risk. Rather than changing production directly, the solution was first validated in a dedicated development environment in order to bring to surface operational challenges and refine the production deployment procedure. The target architecture comprised of the following components:

- SQL Server Enterprise Edition.
- Windows Server Failover Cluster (WSFC).
- SQL Server Always On Availability Groups.
- Manual database seeding for controlled synchronization of large databases.
- Asynchronous-commit availability mode.
- AWS EC2 infrastructure.
- GP3 EBS storage volumes.

High Availability & Read-Replica Implementation

Security

The implemented architecture follows enterprise SQL Server security best practices:

- Windows Authentication.
- SQL Server Enterprise security features.
- Secure HADR endpoints.
- Windows Server Failover Cluster authentication.
- Principle of least privilege.
- Controlled read-only access to replica databases.

Monitoring & Validation

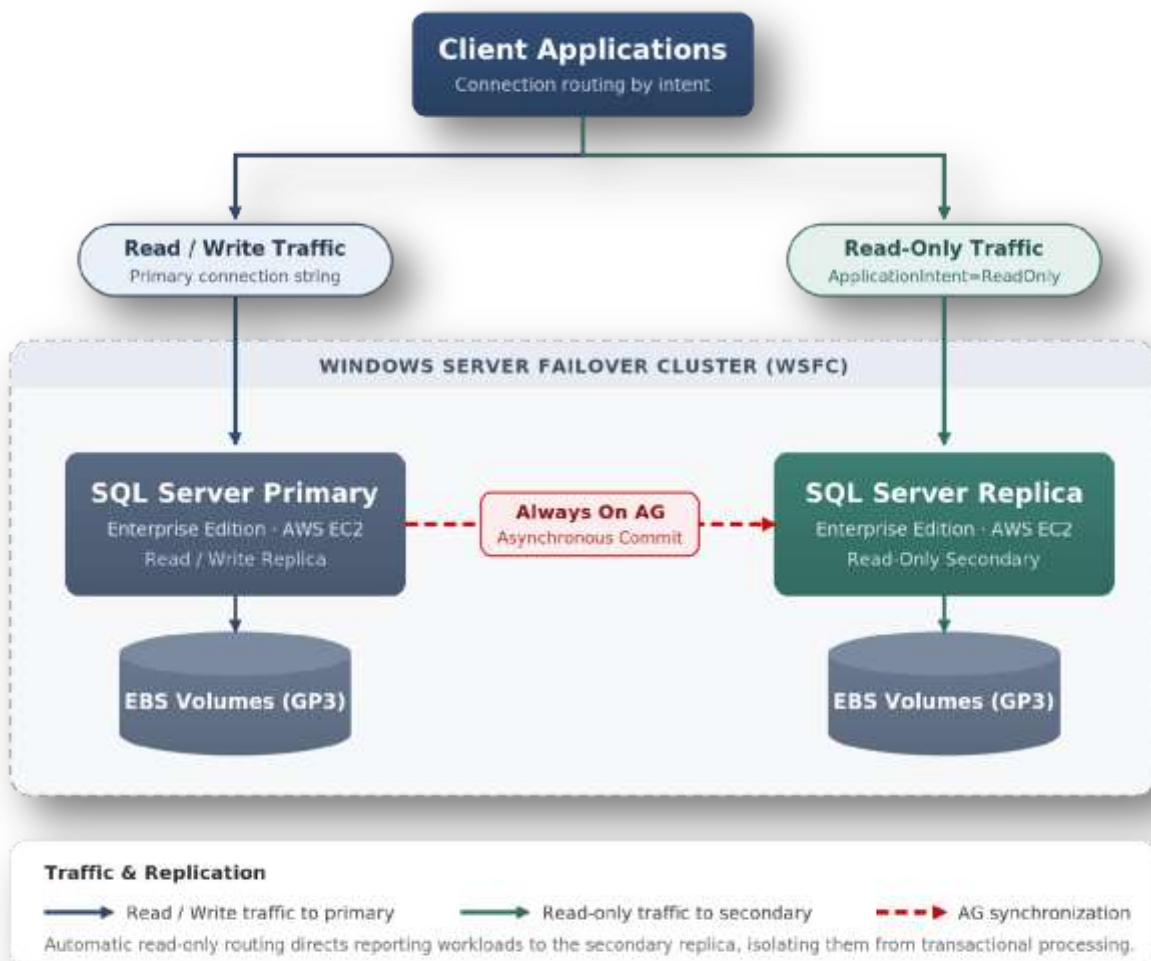
Comprehensive validation was performed throughout implementation to ensure a stable, reliable deployment. Monitoring activities included:

- Availability Group health monitoring.
- Replica synchronization status.
- Database synchronization state.
- SQL Server Agent monitoring.
- Windows Failover Cluster validation.
- SQL Server error-log monitoring.
- Read-only connectivity validation.

High Availability & Read-Replica Implementation

Architecture:

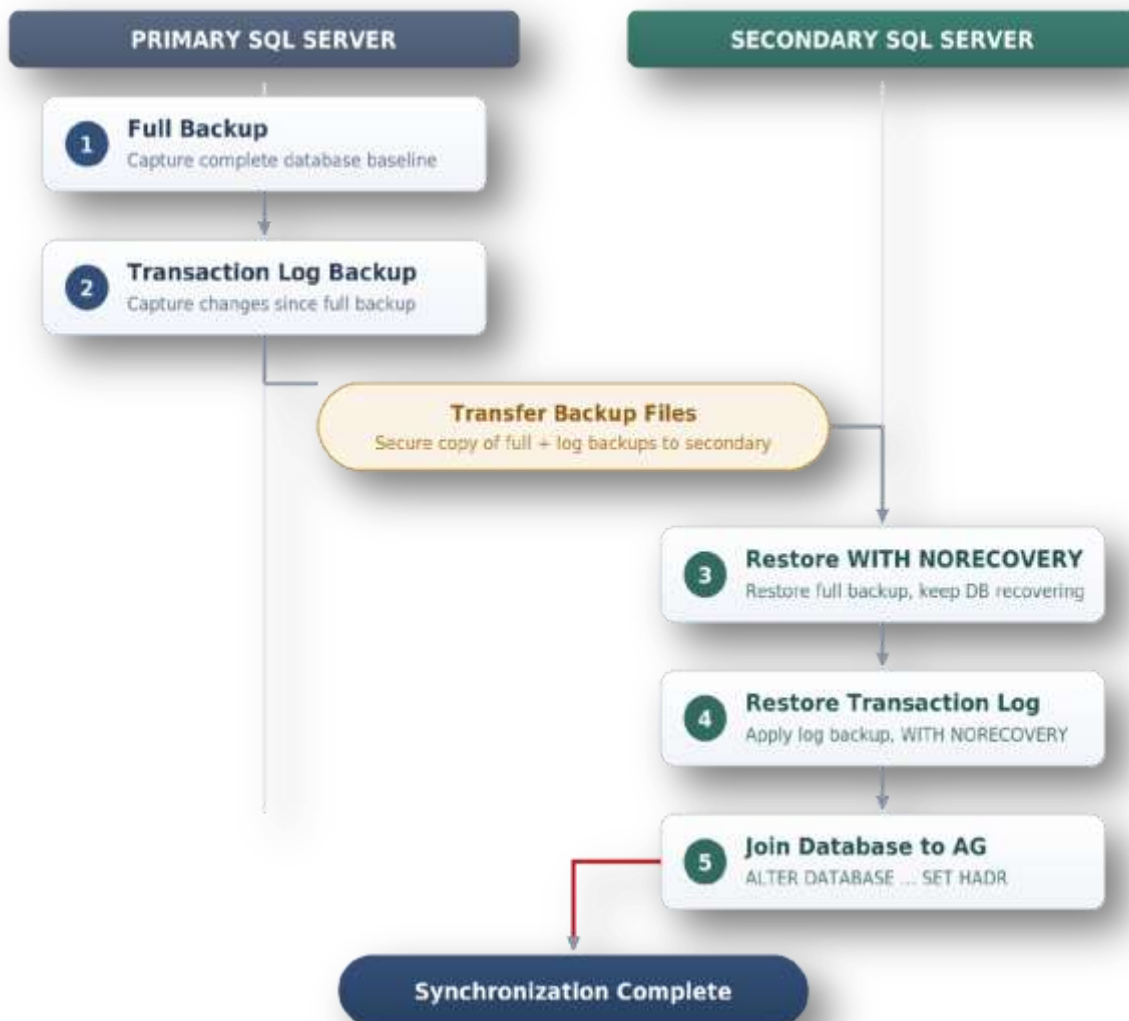
A secondary replica was introduced within an Always On Availability Group (AG) on a Windows Server Failover Cluster. Applications route read/write traffic to the primary replica, while read-only reporting traffic is directed to the secondary replica. Data is kept current across replicas through asynchronous commit synchronization.



High Availability & Read-Replica Implementation

Database Seeding

To support large databases while retaining full operational control, manual database seeding was selected over automatic seeding. Each database was added to the Availability Group via a deterministic backup-and-restore sequence, allowing the team to control timing, validate each step, and batch the workload by database size.



High Availability & Read-Replica Implementation

Benefits:

Following implementation, the Client realized a set of strategic and operational benefits:

- Enterprise-grade SQL Server high-availability architecture.
- A dedicated read-only reporting replica.
- Reduced reporting workload on the production server.
- Improved operational resilience.
- A scalable architecture supporting future replicas.
- Lower implementation risk through phased validation.
- A standardized, repeatable deployment process.
- Improved database administration and monitoring capabilities.

Conclusion:

The engagement successfully established enterprise-grade SQL Server Always On Availability Group architecture for the client, delivering a dedicated read-only replica while maintaining production stability. By validating the implementation in a development environment before production rollout, the team minimized deployment risk, resolved key engineering challenges in advance, and produced a repeatable implementation approach for future scalability.

The resulting architecture provides a strong foundation for enhanced availability, improved reporting performance, and future expansion of the Client's SQL Server infrastructure.