Pre-Implementation (Part 1)

GenAI for SQL Server to Aurora PostgreSQL Migration - AWS DMS Schema Conversion with GenAI

Use Case Lead: Use Case Lead: Database Team Lead

Contributors: Contributors: Database Team, DevOps, Migration Team

Reviewers: Reviewers: [TBD]

Overview

Description:

- Migration from SQL Server to Aurora PostgreSQL requires comprehensive schema conversion, including tables, views, stored procedures, functions, and triggers
- AWS DMS Schema Conversion with Generative AI (launched Dec 2024) is purpose-built for this exact migration path
- Converts up to 90% of schema automatically, including complex code objects that traditionally require manual conversion

Business need / value:

- Automated conversion of 90% of schema objects vs 50-60% with traditional tools
- Reduces migration timeline by 60-70% compared to manual conversion
- Beta customers achieved significant cost savings through reduced manual effort
- Built-in assessment reports provide migration complexity analysis upfront

Users and applicable business units:

Database Teams, Application Development Teams, DevOps, QA Teams

Cost and funding source:

- Implementation: Pay only for storage used during conversion (minimal cost)
- Ongoing: No additional GenAI costs - uses AWS Bedrock in backend
- DMS migration: Pay by the hour for capacity used during data migration
- AWS Free Tier: $100 credits available for new users

Solution Plan

1. Inventory of data assets, including classification and sensitivity levels:

Input: SQL Server database schemas, stored procedures, functions, triggers, views, DDL scripts, data types, constraints, indexes. Source can be self-managed SQL Server on EC2, on-premises, or RDS SQL Server

2. Proposed controls, such as encryption, access control etc:

Access controlled through AWS IAM policies, data encrypted in transit and at rest, audit logging via CloudTrail, Secrets Manager for credentials, S3 bucket encryption for assessment reports

3. Alignment with reference architectures:

AWS DMS is a native AWS managed service following AWS Well-Architected Framework. Fully integrated with AWS ecosystem (VPC, IAM, KMS, CloudWatch). Uses Amazon Bedrock for GenAI capabilities ensuring compliance with AWS AI governance

4. Data collection, storage, and processing:

DMS Schema Conversion reads source database metadata, generates assessment reports stored in S3, converts schema using combination of rule-based engine + GenAI for complex objects. No customer data sent to GenAI models - only schema/code structures

5. Details of the AI models, including architecture, deployment methods:

Model: Large Language Models (LLMs) via Amazon Bedrock. Method: Hybrid approach - rule-based conversion for standard objects, GenAI for complex procedures/functions. Architecture: Fully managed, serverless. Objects converted: SELECT, INSERT, UPDATE, DELETE, MERGE statements, stored procedures, functions, triggers, data type conversions, system function equivalents. Scope: Converts 90% of schema automatically vs 50-60% traditional rule-based only

Legal and Compliance and InfoSec

1. Approach to ensure InfoSec (confidentiality, integrity, availability) of AI systems and data:

AWS managed service with enterprise-grade security. VPC isolation, encryption at rest/transit, IAM role-based access. GenAI feature is opt-in - can be disabled for strict compliance requirements. Schema metadata only used for conversion - no actual customer data processed by AI

2. Financial and tech regulatory requirements:

Compliant with SOC 2, ISO 27001, GDPR, HIPAA eligible. AWS DMS inherits all AWS compliance certifications. Assessment reports can be exported for audit purposes

3. Approach to understand oversight - ensure existing controls can comply with regulations:

Built-in AWS governance through Control Tower, Config, and Security Hub. Conversion results are reviewable before application. Action items clearly marked for items requiring manual review. Audit trail maintained via CloudTrail

4. VRAP outcome (if required):

Low-Medium risk - AWS managed service with opt-in GenAI. Can operate without GenAI if required by policy

Controls

Documentation of the controls (internal/external) to be implemented and how they will be enforced and tested:

- All GenAI-converted objects clearly marked with 'Generated by AI' tags in assessment reports
- Converted code must be reviewed by database architects before deployment to production
- GenAI feature can be disabled organization-wide for compliance requirements
- Automated testing post-conversion to validate schema integrity and functionality
- Version control integration for tracking all schema changes
- Assessment reports stored in encrypted S3 buckets with access logging enabled

Pre-Implementation (Part 1)

GenAI for SQL Server to Aurora PostgreSQL Migration - AWS DMS + Amazon Q Developer

Use Case Lead: Use Case Lead: Database + Development Team Lead

Contributors: Contributors: Database Team, Development Team, DevOps

Reviewers: Reviewers: [TBD]

Overview

Description:

- Combines AWS DMS Schema Conversion (for database schema) with Amazon Q Developer (for application code updates)
- DMS handles schema conversion, Q Developer assists developers in updating application queries and connection strings
- Best for migrations where application code also needs significant refactoring

Business need / value:

- Comprehensive coverage from database schema to application code layer
- Amazon Q Developer provides real-time assistance for SQL query refactoring in IDE
- Reduces developer learning curve for PostgreSQL syntax differences
- Integrated security scanning catches vulnerabilities during code updates

Users and applicable business units:

Database Teams, Application Development Teams, DevOps Engineers

Cost and funding source:

- Implementation: DMS pay-per-use + $19/user/month for Q Developer Professional
- Ongoing: Monthly subscription per developer using Q Developer
- Best ROI when application code requires significant updates alongside schema migration

Solution Plan

1. Inventory of data assets, including classification and sensitivity levels:

Input: Database schemas (handled by DMS), application source code, SQL queries embedded in code, ORM configurations, database connection logic, integration test suites

2. Proposed controls, such as encryption, access control etc:

AWS IAM for DMS access, AWS SSO for Q Developer, code review workflows for Q-generated suggestions, security scanning integrated in CI/CD pipeline

3. Alignment with reference architectures:

Combines two native AWS services for end-to-end migration coverage. DMS for database layer, Q Developer for application layer. Both follow AWS security and compliance standards

4. Data collection, storage, and processing:

DMS processes database metadata only. Q Developer analyzes code within developer's IDE. No code leaves secure environment unless explicitly shared

5. Details of the AI models, including architecture, deployment methods:

DMS: Uses Amazon Bedrock LLMs for schema conversion. Q Developer: AWS proprietary models for code assistance. Combined benefits: Schema converted by DMS, then Q Developer helps update application code with PostgreSQL-specific syntax, query optimization suggestions, connection string updates, ORM configuration changes

Legal and Compliance and InfoSec

1. Approach to ensure InfoSec (confidentiality, integrity, availability) of AI systems and data:

DMS: AWS managed with VPC isolation. Q Developer: Enterprise SSO, code not stored outside organization, built-in AWS security controls

2. Financial and tech regulatory requirements:

Both services inherit AWS compliance certifications (SOC 2, PCI DSS, HIPAA eligible). Q Developer can be configured to exclude sensitive repositories

3. Approach to understand oversight - ensure existing controls can comply with regulations:

Native AWS governance model. Q Developer suggestions require developer approval before implementation. Usage monitoring via CloudWatch

4. VRAP outcome (if required):

Low-Medium risk - Both are AWS managed services with enterprise controls

Controls

Documentation of the controls (internal/external) to be implemented and how they will be enforced and tested:

- DMS conversions reviewable before deployment
- Q Developer suggestions require explicit developer acceptance
- Automated security scanning on all Q-generated code
- Integration with existing code review and CI/CD processes
- Usage telemetry and audit logs maintained for compliance
- Ability to disable Q Developer for specific sensitive projects

Pre-Implementation (Part 1)

GenAI for SQL Server to Aurora PostgreSQL Migration - AWS DMS + Striim (Hybrid Cloud)

Use Case Lead: Use Case Lead: Database Migration Team Lead

Contributors: Contributors: Database Team, Infrastructure Team, DevOps

Reviewers: Reviewers: [TBD]

Overview

Description:

- Uses AWS DMS for schema conversion, Striim for real-time continuous data replication with sub-second latency
- Best for large-scale migrations requiring zero downtime and continuous synchronization
- Striim's CDC (Change Data Capture) technology enables hybrid cloud strategies during migration

Business need / value:

- Zero downtime migration - source and target databases stay synchronized during transition
- Sub-second latency for data replication supports hybrid cloud operation
- Transactional integrity maintained - preserves primary keys, foreign keys, dependencies
- Ideal for large databases (100GB+) requiring minimal business disruption

Users and applicable business units:

Database Teams, Infrastructure Teams, Business Continuity Planning

Cost and funding source:

- Implementation: DMS pay-per-use + Striim enterprise licensing (quote-based)
- Ongoing: Striim subscription based on data volume and connectors
- Higher upfront cost but best ROI for mission-critical, zero-downtime requirements

Solution Plan

1. Inventory of data assets, including classification and sensitivity levels:

Input: SQL Server production databases, transaction logs for CDC, database schemas, large data volumes (100GB to multi-TB), performance baselines, SLA requirements

2. Proposed controls, such as encryption, access control etc:

TLS encryption for data in transit, role-based access control in Striim, AWS IAM for DMS, encryption at rest in Aurora, audit logging in both systems

3. Alignment with reference architectures:

Enterprise-grade architecture: DMS for initial schema conversion, Striim for continuous replication. Supports on-premises to cloud, multi-cloud, and hybrid cloud patterns. Pre-built connectors for SQL Server and Aurora PostgreSQL

4. Data collection, storage, and processing:

DMS converts schema. Striim uses CDC to capture changes from SQL Server transaction logs in real-time, applies transformations, and replicates to Aurora with sub-second latency. Handles schema evolution dynamically

5. Details of the AI models, including architecture, deployment methods:

DMS: Amazon Bedrock LLMs for schema conversion (90% automation). Striim: Automated data mapping and transformation engine (not GenAI-based but highly intelligent rule engine). Combined approach: DMS for schema + code objects, Striim for data replication and synchronization. Deployment: Cloud-native, scales automatically with data volume

Legal and Compliance and InfoSec

1. Approach to ensure InfoSec (confidentiality, integrity, availability) of AI systems and data:

End-to-end encryption, secure VPC connectivity, TLS 1.2+. DMS within AWS security boundary. Striim certified for enterprise security with SOC 2 compliance

2. Financial and tech regulatory requirements:

DMS: AWS compliance certifications. Striim: SOC 2 Type II, ISO 27001 certified. GDPR and HIPAA compliant configurations available

3. Approach to understand oversight - ensure existing controls can comply with regulations:

Comprehensive monitoring via Striim dashboard and AWS CloudWatch. Real-time alerting for replication lag, errors, or anomalies. Audit logs maintained for compliance reviews

4. VRAP outcome (if required):

Medium risk - involves third-party software (Striim) but with enterprise security certifications

Controls

Documentation of the controls (internal/external) to be implemented and how they will be enforced and tested:

- DMS schema conversion reviewed before Striim replication starts
- Striim provides real-time monitoring dashboard for replication health
- Automated rollback capability if data integrity issues detected
- Validation queries run continuously to ensure source-target consistency
- Change log audit trail maintained for all data transformations
- Automated failover and disaster recovery configured

Pre-Implementation (Part 2 - Security)

Security Assessment Summary

Security Lead: [TBD]
Jira Id: TBD
Review Type: Medium
Risk Ranking: Low-Medium

> The proposed use of GenAI for SQL Server to Aurora PostgreSQL migration is dependent on acceptance of AWS DMS Schema Conversion with Generative AI capabilities. Key risks and mitigations are outlined below.

Review Highlights

- AWS DMS Schema Conversion Security Documentation
- GenAI Usage Policy Compliance Review
- Data Classification and Handling Procedures for Migration
- Third-party Tool Assessment (if using Striim)

Risk Issues and Mitigations

Risk Description and Mitigation	Impact	Likelihood
Risk #1: GenAI-converted schema objects (procedures, functions, triggers) may contain logical errors or performance issues not detected during automated conversion, potentially causing application failures post-migration **Mitigation:** All GenAI-converted objects clearly marked for mandatory review by database architects. Comprehensive testing in non-production environment before production deployment. Automated testing suite for regression detection.	High	Medium
Risk #2: Incomplete schema conversion or data type incompatibilities could result in data loss, corruption, or referential integrity violations during migration **Mitigation:** DMS provides detailed assessment reports identifying conversion gaps upfront. Validation queries run pre and post-migration. Use DMS data validation feature. Maintain synchronized source environment during transition period.	High	Low
Risk #3: Database schema metadata processed by GenAI could inadvertently expose proprietary business logic or sensitive database designs to AI model providers **Mitigation:** AWS states only schema structures (not data) used for conversion. GenAI feature is opt-in and can be disabled. AWS Bedrock models in isolated environment. Compliance with AWS data processing agreements.	Medium	Low
Risk #4: Migration downtime longer than anticipated could impact business operations, especially for large databases (500GB+) **Mitigation:** Use Striim for zero-downtime migration if critical. Conduct migration during maintenance windows. Implement phased migration approach. Maintain source database operational during transition.	Medium	Medium
Risk #5: Performance degradation post-migration if Aurora PostgreSQL not properly tuned for converted workload patterns **Mitigation:** Conduct performance testing in staging environment. Use Aurora Performance Insights for monitoring. Engage AWS support for query optimization. Plan for iterative performance tuning phase post-migration.	Medium	Medium