# Solution Context - Scheduled Historical Scanned Image Migration

## Overview

This document provides a high-level solution context for the Scheduled Historical Scanned Image Migration process, which involves selecting document information from an Oracle database at scheduled intervals using Spring Batch, transferring data via API Connect (API-C) over AWS Direct Connect, and storing the documents and metadata in AWS.

## Solution Architecture

### High-Level Flow

• Spring Batch Job (On-Premises) - Runs at scheduled intervals, retrieves document metadata from Oracle, and sends it to API-C.

• API Connect (On-Premises) - Acts as a gateway and routes requests securely via AWS Direct Connect.

• AWS Direct Connect - Ensures a private, high-speed, and secure connection between on-prem and AWS.

• AWS API Gateway - Manages and routes incoming API requests to AWS Lambda.

• AWS Lambda (Create) - Generates a unique GUID, stores document in S3, and saves metadata in DynamoDB.

• Metadata Storage - Amazon S3 stores document content; Amazon DynamoDB stores metadata.

• Returning GUID to On-Prem - GUID is stored back in Oracle for tracking.

## Key Components & Responsibilities

|  |  |
| --- | --- |
| Component | Function(s) |
| Spring Batch (On-Premises) | Retrieves document info from Oracle and triggers API call to API-C. |
| Oracle Database | Stores document metadata before and after migration. |
| API Connect (On-Premises) | Routes API requests securely to AWS API Gateway. |
| AWS Direct Connect | Provides private, high-speed connectivity between on-prem and AWS. |
| AWS API Gateway | Exposes API endpoints for processing document migration requests. |
| AWS Lambda (Create) | Generates GUID, stores documents in S3, and saves metadata in DynamoDB. |
| Amazon S3 | Stores scanned document images securely. |
| Amazon DynamoDB | Stores document metadata (GUID, S3 path, Timestamp). |

## Error Handling & Retry Strategy

|  |  |  |
| --- | --- | --- |
| Component | Error Handling Mechanism | Impact |
| Spring Batch | Implements retry logic and logs failures. | Prevents missing document records. |
| API Connect | Provides response transformation and error logging. | Ensures structured API request forwarding. |
| AWS API Gateway | Returns structured HTTP error codes. | Helps identify API failures easily. |
| AWS Lambda | Implements try/catch blocks with retries on failures. | Reduces failures due to transient issues. |
| Amazon S3 | Manages permission errors and file existence checks. | Prevents unauthorized access issues. |
| Amazon DynamoDB | Handles conditional writes to prevent data corruption. | Ensures consistency in metadata storage. |

## Interfaces & Protocols

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Interface | Source | Target | Protocol | Description |
| 1 | API Request Flow | Spring Batch | API Connect (On-Prem) | HTTPS (REST) | Sends document metadata to API-C. |
| 2 | Secure Connectivity | API Connect | AWS Direct Connect | Private Link | Ensures secure request forwarding. |
| 3 | API Invocation | API Gateway | AWS Lambda | HTTPS (REST) | Routes requests to AWS Lambda. |
| 4 | Document Storage | AWS Lambda (Create) | Amazon S3 | HTTPS (S3 API) | Uploads document files to S3. |
| 5 | Metadata Storage | AWS Lambda (Create) | Amazon DynamoDB | HTTPS (SDK) | Saves document metadata. |
| 6 | GUID Return | AWS Lambda | API Gateway | HTTPS (REST) | Returns generated GUID. |
| 7 | GUID Storage | API Connect | Oracle Database | SQL Query | Updates Oracle with GUID. |

## Technology Stack

|  |  |
| --- | --- |
| Component | Technology |
| Spring Batch | Java, Spring Boot, Spring Batch |
| Oracle Database | Oracle 12c/19c |
| API Connect (On-Premises) | IBM API Connect, Nginx |
| AWS Direct Connect | AWS Direct Connect |
| AWS API Gateway | AWS API Gateway (REST) |
| AWS Lambda | AWS Lambda (Python) |
| Amazon S3 | AWS S3 (Object Storage) |
| Amazon DynamoDB | AWS DynamoDB (On-Demand) |

## Security & Compliance Considerations

• \*\*API Authentication:\*\* API Gateway protected using IAM Roles and JWT authentication.

• \*\*Data Encryption:\*\* In-Transit (HTTPS - TLS 1.2), At-Rest (AWS KMS encryption for S3 and DynamoDB).

• \*\*Access Controls:\*\* IAM roles for API access, S3 bucket policies for restricted access.

• \*\*Logging & Monitoring:\*\* AWS CloudWatch for API Gateway, Lambda, and DynamoDB logs.