# AWS CRUD Document Management System - Design Document

## 1. Overview

This document describes the design and architecture of a serverless document management system that performs CRUD (Create, Read, Update, Delete) operations using AWS services.

## 2. Architecture Components

|  |  |
| --- | --- |
| AWS Service | Purpose |
| API Gateway | Exposes HTTP endpoints for each CRUD operation. |
| AWS Lambda | Executes business logic for Create, Read, Update, Delete (each operation has a separate function). |
| Amazon S3 | Stores actual document files securely. |
| Amazon DynamoDB | Stores metadata (GUID, S3 path, timestamps, etc.). |
| AWS Direct Connect | Ensures a secure and low-latency connection from on-premise services. |

## 3. High-Level Architecture

1. Client calls API Gateway with CRUD requests.  
2. API Gateway routes request to the respective Lambda function.  
3. Lambda function processes the request, interacting with S3 (for document storage) and DynamoDB (for metadata storage).  
4. Lambda returns a GUID for reference and future queries.

## 4. API Design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Operation | HTTP Method | Endpoint | Lambda Function | Description |
| Create | POST | /documents | CreateDocumentLambda | Uploads a document to S3, stores metadata in DynamoDB, and returns GUID. |
| Read | GET | /documents/{guid} | ReadDocumentLambda | Retrieves document metadata from DynamoDB and generates a pre-signed S3 URL. |
| Update | PUT | /documents/{guid} | UpdateDocumentLambda | Updates an existing document in S3 and metadata in DynamoDB. |
| Delete | DELETE | /documents/{guid} | DeleteDocumentLambda | Removes the document from S3 and its metadata from DynamoDB. |

## 6. Security Measures

✅ IAM Policies restrict access to only authorized resources.  
✅ S3 Bucket Encryption ensures secure document storage.  
✅ Pre-Signed URLs prevent unauthorized access to documents.  
✅ CloudWatch Logging monitors Lambda and API activity.

## 7. Conclusion

This AWS-based CRUD document management system ensures scalability, security, and performance. Each CRUD operation is handled by a dedicated Lambda function, while documents are stored in S3 and metadata in DynamoDB. The design supports secure API access via API Gateway, enabling seamless document management for external applications.