# Service Documentation

## 1. Service Description

|  |  |
| --- | --- |
| Service Name | Description |
| Upload Document | This service allows applications to upload documents via API-C to AWS, where documents are stored in S3, and metadata is stored in DynamoDB. |

## 2. Service Use Cases

|  |  |
| --- | --- |
| Use Case | Description |
| New Document Upload | A new document is uploaded by the user, and a unique GUID is assigned. The document is stored in S3, and its metadata is stored in DynamoDB. |
| Document Replacement | An existing document can be replaced with a new version. The metadata is updated in DynamoDB with the new timestamp and S3 path. |
| Failed Upload Handling | If an upload fails, retry mechanisms ensure successful completion. If retries fail, an error is logged for manual intervention. |

## 3. Service Request Catalog Path by Environment

|  |  |
| --- | --- |
| Environment | Catalog Path |
| Development | /dev/documents/upload |
| Testing | /test/documents/upload |
| Staging | /staging/documents/upload |
| Production | /prod/documents/upload |

## 4. Order of Processing for "Create" (Upload Document)

1. 1. Application → Sends an HTTPS request to API-C (on-premises) with document data.
2. 2. API-C (On-Premises) → Forwards the request via Direct Connect to AWS API Gateway.
3. 3. AWS API Gateway → Routes the request to the Upload Lambda function.
4. 4. AWS Lambda (Upload Service):
5. - Generates a GUID for the document.
6. - Stores the document in S3.
7. - Saves metadata (GUID, S3 URL, timestamp) in DynamoDB.
8. 5. Lambda → API Gateway → Returns GUID & S3 URL in response.
9. 6. API Gateway → API-C → Sends the response back to API-C.
10. 7. API-C → Application → Application receives GUID & confirmation.