Runbook

Secure Shield

PII Redaction Data Pipeline

Paris, 6th August 2025

Generated by:
Amazon Web Services EMEA SARL
(France Branch)

Customer:
HealthTech Analytics

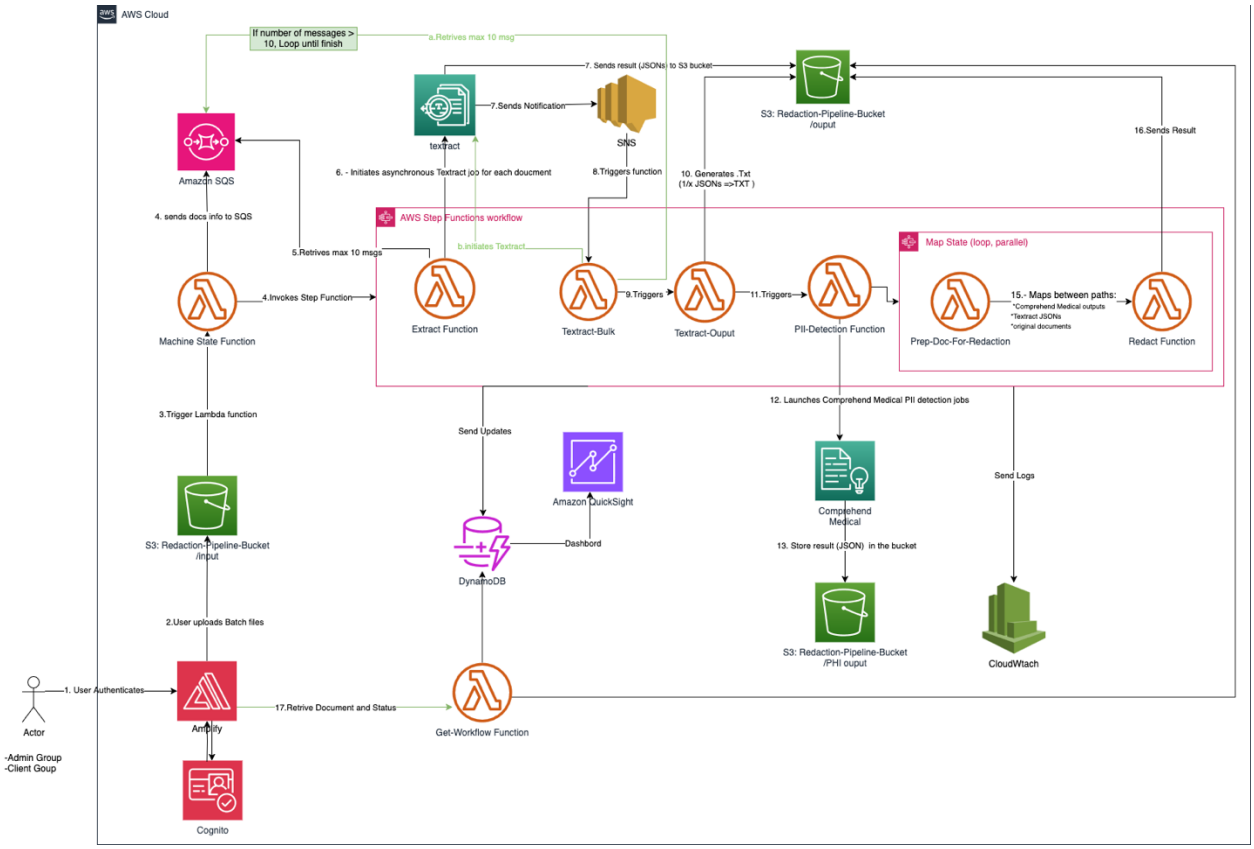
Project member	Company	Role in the project
Miniar Jabri	AWS	Owner

Table of Contents

1. System Overview

Purpose and Architecture

The Secure Shield PII Redaction Pipeline is a serverless solution that automatically detects and redacts Personally Identifiable Information (PII) and Protected Health Information (PHI) from healthcare documents. The solution addresses the critical need for efficient, accurate, and HIPAA-compliant data processing.



Key components:

- Frontend React application hosted on AWS Amplify
- Amazon Cognito for authentication and authorization
- S3 buckets for document storage
- AWS Lambda functions for processing
- Step Functions for workflow orchestration
- Amazon Textract for document text extraction
- Amazon Comprehend Medical for PII/PHI detection
- DynamoDB for workflow status tracking
- Amazon QuickSight for monitoring and analytics

System Requirements

- AWS Account with appropriate permissions
- Supported document formats: PDF, TIFF, PNG, JPEG
- Minimum browser requirements: Chrome (latest), Firefox (latest), Edge (latest)

Data Flow

1. User uploads documents through the web interface
2. Documents are stored in S3
3. Step Functions workflow is triggered
4. Documents are processed through Textract for text extraction
5. Extracted text is analyzed by Comprehend Medical for PHI/PII detection
6. Lambda functions apply redactions based on detected entities
7. Redacted documents are stored in S3
8. User retrieves redacted documents through the web interface

2. Installation and Deployment

Prerequisites

1. AWS Account with administrator permissions
2. AWS CLI installed and configured
3. Node.js (v14+) and npm installed
4. AWS CDK toolkit installed (npm install -g aws-cdk)
5. Python 3.8+ installed

Environment Setup

1. Clone the repository:
2. git clone https://gitlab.aws.dev/miniarja/Pii-Redaction-Pipeline.git
3. cd Pii-Redaction-Pipeline-main
4. Install backend dependencies:
5. cd backend
6. npm install
7. Install frontend dependencies:
8. cd ../frontend

9. npm install
10. Configure environment variables: Create a .env file in the backend directory with the following variables:
11. ROOT_BUCKET=pii-redaction-pipeline-docs-[your-unique-suffix]
12. DOMAIN_COGNITO=pii-redaction-pipeline
13. PII_REGION=us-east-1
14. ADMIN_USER=admin@example.com
15. ADMIN_PASSWORD=[secure-password]
16. CUSTOMER_USER=customer@example.com
17. CUSTOMER_PASSWORD=[secure-password]
18. CUSTOMER_ROLE=customer

Infrastructure Deployment

1. Bootstrap CDK (first time only):
2. cd backend
3. cdk bootstrap aws://[ACCOUNT-NUMBER]/[REGION]
4. Deploy the infrastructure:
5. cdk deploy --all

   This will deploy:

   - CdkPIIAppStack (Cognito, S3)
   - PIIBackendStack (DynamoDB)
   - PIILambdaStack (Lambda functions)
   - PIIStepFunctionStack (Step Functions workflow)
   - PIIWebDeployStack (Amplify hosting)
6. Note the outputs:
   - Cognito UserPool ID
   - Cognito Web Client ID
   - Identity Pool ID
   - S3 Bucket name
   - Amplify application URL

Verification

1. Access the Amplify URL provided in the CDK outputs
2. Login using the admin credentials defined in environment variables

3. Upload a test document to verify the pipeline functionality
4. Monitor the Step Functions execution in the AWS Console

3. User Guide

Authentication

1. Navigate to the application URL
2. Login using your provided credentials:
    - Admin users: email and password provided by administrator
    - Customer users: email and password provided by administrator
3. No self-signup is enabled - all users must be created by administrators

Document Upload

1. Navigate to "Process Documents" in the main navigation
2. Click "Upload Documents"
3. Select files (supported formats: PDF, TIFF, PNG, JPEG)
4. Click "Upload" - a workflow ID will be assigned
5. The system supports batch uploads (up to 200 documents at once)

Workflow Monitoring

1. Navigate to "Review Documents" in the main navigation
2. View the list of workflows with their status:
    - Submitted: Documents uploaded, waiting for processing
    - Processing: Documents currently being processed
    - Completed: All documents have been processed
    - Failed: One or more documents failed to process
3. Click on a workflow to view details

Viewing Results

1. In the workflow details page, select the "PHI" tab
2. Select a document from the list
3. The document viewer will show:
    - For Admin users: Original document with PHI entities highlighted and listed
    - For Customer users: Redacted document with PHI information masked

Document Download

1. When viewing a document, click the download button
2. The document will download to your local machine:
    - Admin users can download both original and redacted versions
    - Customer users can download only redacted versions

4. Administrative Guide

User Management

1. Access AWS Console and navigate to Amazon Cognito
2. Select the user pool (name format: pii-redaction-pipeline-userpool)
3. Add new users:
    - Click "Create user"
    - Enter email, name, and temporary password
    - Select "Generate a password" or enter a custom password
    - Click "Create user"
4. Assign users to groups:
    - Select the user from the list
    - Click "Add to group"
    - Select either "admin" or "customer" group
    - Click "Add to group"

System Monitoring

1. **CloudWatch Logs**:
    - Navigate to CloudWatch in AWS Console
    - Check Log Groups for each Lambda function:
        - /aws/lambda/workflow-state-machine-*
        - /aws/lambda/pii-detection
        - /aws/lambda/redact
        - etc.
    - Set up log filters for error monitoring
2. **Step Functions Dashboard**:
    - Navigate to Step Functions in AWS Console
    - Select workflow-state-machine
    - View execution history and details
3. **QuickSight Dashboard**:
    - Access QuickSight dashboard for high-level metrics:
        - Document processing volumes
        - Processing times
        - Success/failure rates
        - PHI entity detection statistics

Performance Monitoring

1. Create CloudWatch alarms for:
    - Lambda execution errors
    - Step Function execution failures
    - DynamoDB throttling events
    - SQS queue depth exceeding thresholds
2. Monitor QuickSight dashboards for:
    - Document processing times
    - Textract and Comprehend Medical job durations
    - Redaction processing times
    - Overall workflow completion times

5. Maintenance and Operations

Routine Maintenance

1. **S3 Bucket Management**:
    - Review lifecycle rules periodically
    - Monitor storage usage and adjust retention policies if needed
    - Check for orphaned files (files with no associated workflow)
2. **Database Maintenance**:
    - Monitor DynamoDB capacity usage
    - Consider periodic archiving of old workflow data
    - Check for stale workflow entries (status "processing" but inactive)
3. **Lambda Function Updates**:
    - Apply security patches and library updates
    - Monitor function performance and adjust memory allocations
    - Review timeout settings for optimal performance

Scaling Considerations

1. **Lambda Concurrency**:
    - Monitor concurrent execution metrics
    - Adjust reserved concurrency for critical functions:
        - Document processing: 100 concurrent executions
        - PII detector: 100 concurrent executions
        - Redaction handler: 100 concurrent executions
2. **Step Functions**:
    - Monitor Map State parallel execution counts
    - Default limit: 40 parallel map iterations (configurable)
    - Contact AWS Support for limit increases if needed
3. **SQS Queue**:
    - Monitor queue depth and processing times

- Adjust visibility timeout if jobs are being processed multiple times
- Configure dead-letter queue for failed message handling

Backup and Recovery

1. **Document Backup**:
   - S3 versioning is enabled by default
   - Enable cross-region replication for disaster recovery
   - Implement regular S3 bucket backup procedures
2. **Database Backup**:
   - Enable Point-in-time recovery for DynamoDB tables
   - Schedule regular backups using AWS Backup
   - Test recovery procedures periodically

Update Procedures

1. **Frontend Updates**:
2. cd frontend
3. npm install # Update dependencies
4. npm run build
5. # Amplify will automatically deploy from the connected repository
6. **Backend Updates**:
7. cd backend
8. npm install # Update dependencies
9. cdk deploy --all # Deploy updated infrastructure
10. **Lambda Function Updates**:
    - Update code in /backend/src/lambda/
    - Deploy using CDK or direct upload to Lambda console
    - Test with sample documents before full deployment

6. Troubleshooting

Common Issues and Solutions

1. **Document Upload Failures**:
   - **Symptom**: Document fails to upload
   - **Possible causes**:

        - Document exceeds size limit (100MB)
        - Invalid file format
        - S3 permissions issues
    - **Solutions**:
        - Check file size and format
        - Verify Cognito authentication is working
        - Check IAM roles and S3 bucket policies
2. **Processing Stuck in "Processing" State**:
    - **Symptom**: Workflow status remains "processing" for extended periods
    - **Possible causes**:
        - Step Functions execution failure
        - Lambda function timeout
        - Textract or Comprehend Medical service issues
    - **Solutions**:
        - Check Step Functions execution status in AWS Console
        - Review CloudWatch logs for Lambda functions
        - Check service health dashboards
        - For recovery, consider resubmitting the workflow
3. **PHI Detection Issues**:
    - **Symptom**: PHI entities not detected or incorrectly detected
    - **Possible causes**:
        - Poor document quality
        - Unsupported document format
        - Textract extraction failure
    - **Solutions**:
        - Check Textract output JSON for correct text extraction
        - Review Comprehend Medical response for entity detection
        - Improve document quality if possible
        - Consider manual review for critical documents
4. **Redaction Quality Issues**:
    - **Symptom**: Redacted document has missed PHI or over-redacted content
    - **Possible causes**:
        - Textract bounding box inaccuracy
        - PHI detection confidence threshold issues
        - Document formatting challenges
    - **Solutions**:
        - Check redaction function logs for matching issues
        - Adjust confidence thresholds for PHI detection
        - Review document format and consider preprocessing

Error Codes and Interpretation

1. **Lambda Function Error Codes**:
    - ERR_TEXTRACT_FAILED: Textract processing failed
        - Check document format and quality

        - Verify IAM permissions for Textract service
    - ERR_PHI_DETECTION_FAILED: Comprehend Medical processing failed
        - Check input text format
        - Verify IAM permissions for Comprehend Medical
    - ERR_REDACTION_FAILED: Document redaction failed
        - Check document format compatibility
        - Verify sufficient Lambda memory and timeout settings
2. **Step Functions Error Codes**:
    - States.Timeout: State machine execution timed out
        - Check Lambda function timeout settings
        - Review Textract/Comprehend job durations
    - Lambda.Unknown: Lambda function returned an unhandled error
        - Check CloudWatch logs for detailed error information
    - States.TaskFailed: Task state failed during execution
        - Investigate specific state failure in Step Functions execution history

Log Locations and Analysis

1. **Lambda Function Logs**:
    - CloudWatch Log Groups:
        - /aws/lambda/pii-detection
        - /aws/lambda/redact
        - /aws/lambda/extract
        - /aws/lambda/textract-output
        - etc.
    - Key log patterns:
        - ERROR: Indicates a function error
        - WARNING: Indicates potential issues
        - DEBUG: Detailed processing information
2. **Step Functions Execution Logs**:
    - AWS Step Functions console
    - View execution history for detailed state transitions
    - Check input/output for each state
    - Review error details for failed executions
3. **Frontend Application Logs**:
    - Browser console logs
    - CloudWatch Logs for Amplify hosting
    - S3 access logs for document access patterns

Escalation Procedures

1. **Level 1 Support**:
    - Basic troubleshooting using this guide
    - Review logs and error messages

    - Restart workflows if necessary
2. **Level 2 Support**:
    - Detailed log analysis
    - AWS Console investigation
    - Service limit adjustments
    - Configuration changes
3. **Level 3 Support**:
    - Code modifications
    - Infrastructure updates
    - AWS Support engagement
    - Architecture review

7. Security Management

Access Control Best Practices

1. **IAM Roles and Policies**:
    - Follow least privilege principle
    - Regularly audit IAM permissions
    - Use IAM Access Analyzer to identify unused permissions
    - Implement resource-level permissions where possible
2. **Cognito User Management**:
    - Enforce strong password policies
    - Implement MFA for administrator accounts
    - Regularly audit user accounts and permissions
    - Remove inactive users promptly
3. **Role-Based Access Control**:
    - Maintain clear separation between admin and customer roles
    - Regularly review group memberships
    - Document approval processes for role changes

Data Protection

1. **Encryption**:
    - S3 server-side encryption (SSE-S3) is enabled for all buckets
    - Data in transit protected via HTTPS/TLS
    - Consider using KMS customer managed keys for enhanced control
2. **Data Lifecycle**:
    - Configure appropriate retention policies for PII/PHI data
    - Implement lifecycle rules for long-term storage
    - Document data deletion procedures
3. **Document Access**:
    - Use pre-signed URLs with short expiration times
    - Log all document access events

- Implement IP-based restrictions for sensitive documents

Audit Logging

1. **Enable CloudTrail**:
   - AWS API call logging enabled
   - Object-level logging for S3 buckets
   - Log file validation enabled
   - Multi-region logging considered for global services
2. **Log Monitoring**:
   - Configure CloudWatch Logs retention periods
   - Set up metric filters for security events
   - Create alarms for suspicious activities
3. **Access Reviews**:
   - Schedule regular access reviews
   - Document review process and findings
   - Implement remediation procedures for identified issues

HIPAA Compliance

1. **Data Handling**:
   - PHI/PII data always encrypted at rest and in transit
   - Clear access controls for PHI data
   - Role-based access enforcement in frontend and backend
2. **Audit Controls**:
   - Comprehensive logging of all PHI access
   - Regular log review procedures
   - Incident response plan documented
3. **Technical Safeguards**:
   - Authentication and authorization controls
   - Automatic session timeout
   - Secure document transmission

8. Reference

AWS Resources

1. **S3 Buckets**:
   - Root bucket: [ROOT_BUCKET]
   - Directory structure:
     - /public/input/[workflow_id]/ - Original documents
     - /public/output/[workflow_id]/ - Processed documents

        - /public/phi-output/[workflow_id]/ - PHI detection results
2. **DynamoDB Tables**:
    - Workflow table: workflow-table
        - Primary key: part_key (workflow ID)
        - Sort key: sort_key (document path)
        - Attributes:
            - redaction_status (status of the workflow)
            - timestamp (creation time)
            - documents (list of processed documents)
3. **Lambda Functions**:
    - machine-state - Triggers workflow and manages document batch processing
    - extract - Processes documents with Amazon Textract
    - textract-output - Processes Textract results
    - pii-detection - Submits documents to Amazon Comprehend Medical
    - prep-doc-for-redaction - Maps document paths for redaction
    - redact - Applies redactions to documents
    - get-workflows - Retrieves workflow status information

API Endpoints

1. **Authentication API**:
    - Cognito endpoints for authentication
    - Token-based API access
2. **Document Processing API**:
    - Workflow creation: POST /workflow
    - Workflow status: GET /workflow/{id}
    - Document upload: Pre-signed S3 URLs

File Locations and Naming Conventions

1. **Document Storage**:
    - Original documents: public/input/{workflow_id}/{document_name}
    - Textract results: public/textract/{workflow_id}/{document_name}.json
    - PHI detection results: public/phi-output/{workflow_id}/{document_name}.json
    - Redacted documents: public/output/{workflow_id}/{document_name}-redacted.{extension}
2. **Lambda Function Code**:
    - Main functions: backend/src/lambda/
    - Helper libraries: backend/src/lambda/ (shared modules)
    - Infrastructure code: backend/lib/ (CDK stacks)

Environment Variables

1. **Backend Environment Variables**:

    - ROOT_BUCKET - Main S3 bucket name
    - DOMAIN_COGNITO - Cognito domain prefix
    - PII_REGION - AWS region for deployment
    - ADMIN_USER - Admin username for Cognito
    - ADMIN_PASSWORD - Admin password for Cognito
    - CUSTOMER_USER - Customer username for Cognito
    - CUSTOMER_PASSWORD - Customer password for Cognito
2. **Lambda Environment Variables**:
    - LOG_LEVEL - Logging level (DEBUG, INFO, WARNING, ERROR)
    - PII_TABLE - DynamoDB table name
    - INPUT_BKT - S3 input bucket name
    - SNS_TOPIC - SNS topic ARN for Textract notifications
    - SNS_ROLE - IAM role ARN for SNS publishing
    - IAM_ROLE - IAM role for Comprehend Medical

Performance Limits

1. **System Limits**:
    - Maximum document size: 100MB
    - Maximum batch size: 200 documents
    - Maximum concurrent Lambda executions: 1000
    - Maximum Step Functions Map state concurrency: 40 parallel executions
    - SQS maximum receive count: 10 messages per request
2. **Processing Guidelines**:
    - Small documents (<5MB): <2 minutes processing time
    - Medium documents (5-20MB): <5 minutes processing time
    - Large documents (20-100MB): <15 minutes processing time
    - Maximum daily throughput: 5000+ documents