

# □ Automated Receipt Processing System

## 🌀 Project Vision

Transform the tedious, manual receipt handling process into an intelligent, automated pipeline that **extracts**, **processes**, and **organizes** receipt data with enterprise-grade reliability and scalability.

## 💡 The Challenge We Solve

### Traditional Receipt Management:

- ✗ Manual data entry is slow and error-prone
- ✗ Physical receipts get lost or damaged
- ✗ Difficult to search and retrieve historical data
- ✗ No automated compliance tracking
- ✗ Scaling requires hiring more staff

### Our Automated Solution:

- ✓ **Instant Processing** - Upload and extract data in seconds
- ✓ **99%+ Accuracy** - AI-powered text recognition
- ✓ **Searchable Database** - Find any receipt instantly
- ✓ **Real-time Notifications** - Get alerts as receipts are processed
- ✓ **Infinite Scale** - Handle thousands of receipts simultaneously

## Multi-Layer Architecture Design:

### 📁 Storage Foundation

#### Amazon S3 - *The Digital Vault*

- Secure, durable storage for receipt images & PDFs
- Automatic versioning and backup
- Cost-effective archival for compliance

### ⚡ Event-Driven Processing

#### AWS Lambda - *The Automation Engine*

- Serverless compute triggers on file upload
- Auto-scaling from 1 to 1000+ concurrent executions

- Pay-only-for-usage cost model

## **AI-Powered Extraction**

### **Amazon Textract** - *The Digital Eye*

- Advanced OCR with 99%+ accuracy
- Understands receipt structure (vendor, date, items, totals)
- Handles multiple formats and languages

## **Structured Data Storage**

### **Amazon DynamoDB** - *The Smart Database*

- NoSQL database optimized for fast queries
- Automatic scaling and high availability
- Indexed search across all receipt fields

## **Intelligent Notifications**

### **Amazon SES** - *The Communication Hub*

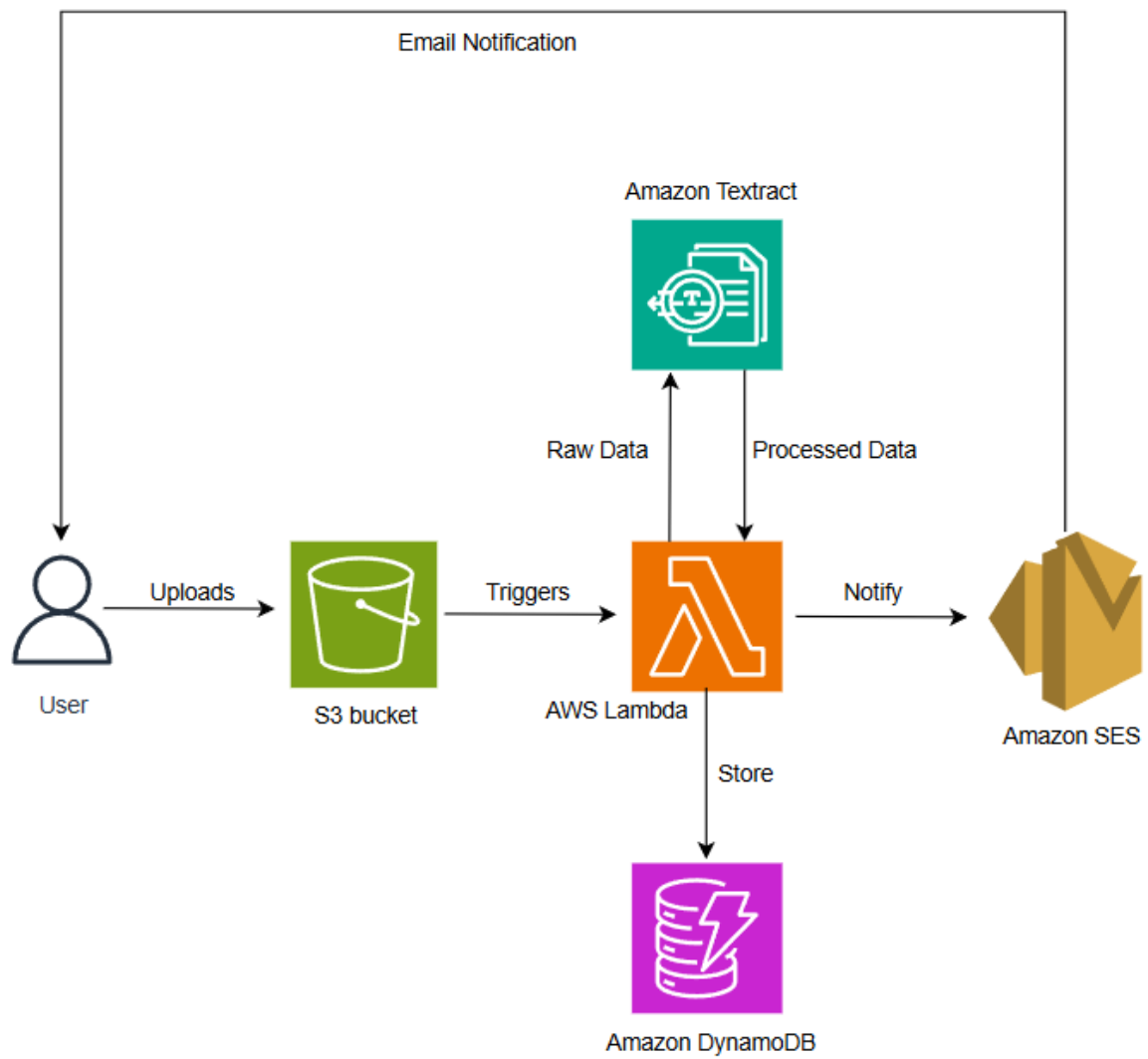
- Professional HTML email reports
- Real-time processing alerts
- Customizable notification rules

## **Enterprise Security**

### **AWS IAM** - *The Security Guardian*

- Principle of least privilege access
- Role-based service permissions
- Audit trails for compliance

## System Architecture

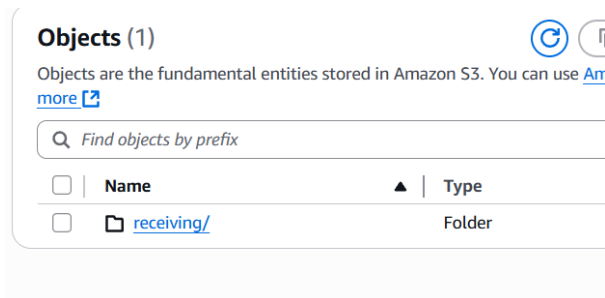


## ☁ AWS Implementation Steps - Completed

### Step 1: Storage Setup

#### S3 Bucket Configuration

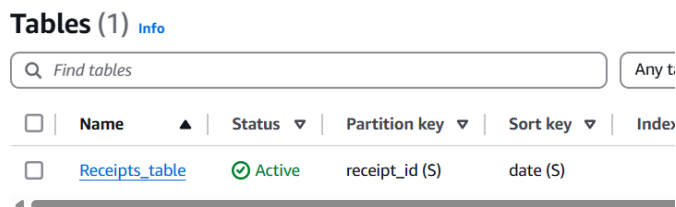
- Created S3 bucket with a dedicated "receiving" folder for incoming files
- Configured bucket for receipt document storage



### Step 2: Database Setup

#### DynamoDB Table Creation

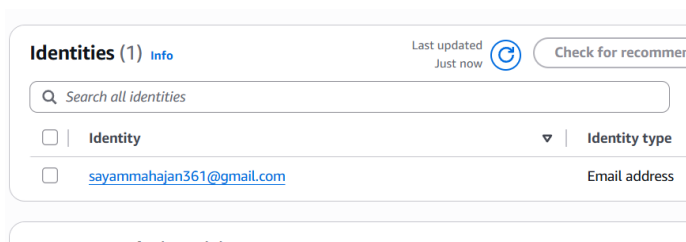
- Set up DynamoDB table with receipt\_id as partition key
- Configured date as sort key for efficient querying and data organization



### Step 3: Email Service Setup

#### SES Identity Verification

- Created and verified email identity in Amazon SES
- Configured email service for sending notifications








### Step 4: IAM Role Configuration

## Lambda Execution Role Setup

- Created IAM role with comprehensive permissions:
  - S3 Read Access (for retrieving uploaded files)
  - DynamoDB Full Access (for database operations)
  - SES Full Access (for email notifications)
  - Textract Full Access (for document text extraction)
  - Lambda Basic Execution (for CloudWatch logging)

Permissions page - Manage permissions

Filter by Type: All types

<input type="checkbox"/>	Policy name <a href="#">?</a>	Type	Attach
<input type="checkbox"/>	 <a href="#">AmazonDynamoDBFullAccess</a>	AWS managed	<a href="#">1</a>
<input type="checkbox"/>	 <a href="#">AmazonS3ReadOnlyAccess</a>	AWS managed	<a href="#">1</a>
<input type="checkbox"/>	 <a href="#">AmazonSESEFullAccess</a>	AWS managed	<a href="#">1</a>
<input type="checkbox"/>	 <a href="#">AmazonTextractFullAccess</a>	AWS managed	<a href="#">1</a>
<input type="checkbox"/>	 <a href="#">AWSLambdaBasicExecutionRole</a>	AWS managed	<a href="#">1</a>

Permissions boundaries (not set)

## Step 5: Lambda Function Creation


### Core Processing Function


- Created Lambda function to handle all project operations
- Attached the configured IAM role for proper permissions
- Implemented business logic for receipt processing
- For Code: Check GitHub repository [Link](#)

### ReceiptLambdaFunc

▼ Function overview [Info](#)

Diagram | Template

 **ReceiptLambdaFunc**

 Layers (0)

### Environment variables (3)

The environment variables below are encrypted at rest with the default Lambda service key.

Q Find environment variables

Key	Value
DYNAMODB_TABLE	Receipts_table
SES_RECIPIENT_EMAIL	sayammahajan361@gmail.com
SES_SENDER_EMAIL	sayammahajan361@gmail.com

## Step 6: Event Integration

### S3 Event Notification Setup

- Configured S3 bucket event notifications
- Linked S3 bucket to Lambda function for automatic triggering
- Enabled real-time processing when files are uploaded

No data events to display.

[Configure in CloudTrail](#)

#### Event notifications (1)

Send a notification when specific events occur in your bucket. [Learn more](#)

<input type="checkbox"/>	Name	Event types	Filters	Destination type	Destination
<input type="checkbox"/>	ReceiptUpload	All object create events	receiving/	Lambda function	<a href="#">ReceiptLambdaFunc</a>

#### Amazon EventBridge

For additional capabilities, use Amazon EventBridge to build event-driven applications at scale using S3 event notifications. [Learn more](#) or [see EventBridge pricing](#)

Send notifications to Amazon EventBridge for all events in this bucket

Edit