



# Secure Shield

## PII Redaction Pipeline

Miniar Jabri (she/her)

Professional Services Consultant Intern @AWS

08/06/2024

# Agenda



## 1. About me!

---



## 2. Context

---

- Customer
- Customer's challenge
- Our solution



## 3. Dive Deep

---

- Technology Stack
- Architecture
- Frontend
- Step function workflow:
  - Text Extraction
  - PII/PHI Detection
  - Redaction
- Monitoring and Observability
- Security
- Limitations
- Next steps



## 4. Lessons Learned

---

# About Me !

# About me

- I'm Miniar, (Meen - yaar) a first time Amazonian
- School: 2<sup>nd</sup> year engineering student at UPSSITECH in Toulouse.
- I graduate in august 2026

During my AWSome Time here....

- I passed the SAA Certificate
- I passed the AI practionier Certificate
- I completed a Gen-AI course on Machine Learning University
- Participated to an instructive firehose on security
- Won the second place in a JAM
- I joined @women\_in\_amazon Group



# Our Customers

# HealthTech Analytics

A HEALTHCARE DATA ANALYTICS COMPANY

*Customer Processes large volumes of patient records and healthcare data to deliver insights for improving patient care and operational efficiency.*

Data Analytics

Healthcare

HIPPA Compliance

Protect PII/PHI



xtelligent  
Healthtech  
Analytics

## Goal

Protect Sensitive information while maintaining data utility for analysis

## WHY IT MATTERS for AWS (Global Context)

The global Healthcare & Life spends \$23.6 billion annually on data analytics, with expected CAGR of 21.4% until 2030.  
=> This highlights the significant market opportunity for building scalable solution

### Source

<https://w.amazon.com/bin/view/APNServiceAcceleration/AnalyticsHome/Programs/ASAP/PLHCLSSolutions/>

# The Problem / Challenge

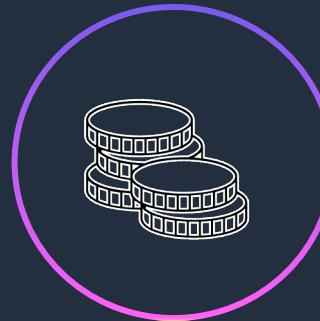
# Current Challenge



Time-consuming: 30-45 minutes per document



Error prone: Human error rate: up to 7%



Expensive: Dedicated staff for manual review  
- Labor cost: \$25-35/Hour  
- Monthly cost (1,000 documents): \$18,000

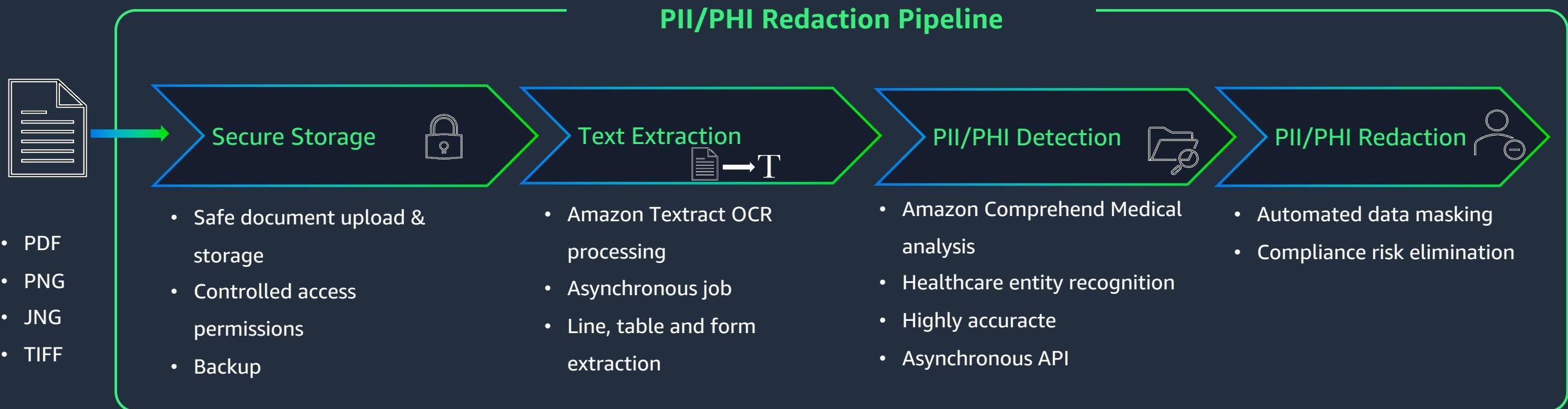
*Manual Process ➔ Processing Delays ➔ Delayed Analytics ➔ Slower Decision Making ➔ Revenue Loss*

**Faster document processing shortens decision cycles; this enables you to serve customer and have people do higher value task**

# Our Solution

# Our Solution - High-level Workflow

How it works:



# PII Redaction Data Pipeline. - Business Value



AWS makes Redaction better, faster, and more cost-effective.



## Simple Operation

- Quick setup with IaC
- Pre-packaged solution that accelerates deployment and saves time

Enables **One-click deployment.**



## Cost-Effective

- Cuts costs by reducing manual labor and errors.
- Pay-per-use AWS Services

Reduces costs by **70%** compared to manual redaction, **\$0.04/page** and **\$450/month** (1000 documents)



## Batch Processing

- Processes thousands of documents per minute.
- Optimized for businesses of any size.

Process **up to 400 Documents** simultaneously with auto-scaling and takes up to **8 minutes.**



## Intuitive User-Interface

- Intuitive dashboard for document upload
- Real-time workflow status tracking.

**Streamline User Experience** with secure document handling."

**“ Privacy is not negotiable. In healthcare, it's the foundation of the doctor-patient relationship and the cornerstone of quality care. ”**

**Kathleen Sebelius**

Former U.S. Secretary of Health and Human Services

# Dive Deep...

# Tech Stack

## Frontend:

- React



## Backend:

- AWS CDK (IaC)
- Javascript (cdk)
- Python (Lambda)

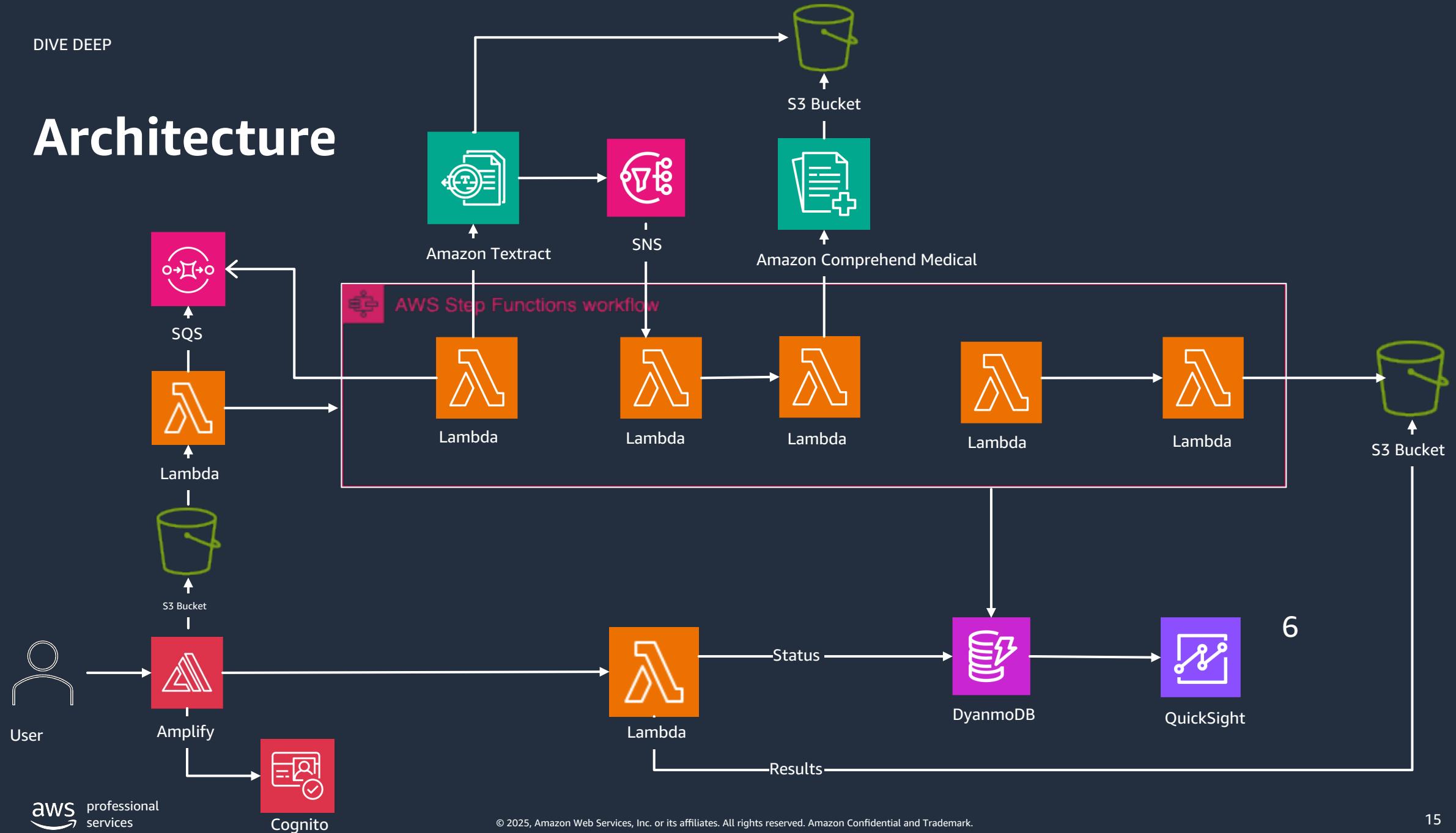


## Version Control:

- GitLab:



# Architecture



# Frontend – User Interface

# User Interface

**Seamless connection  
to AWS services and  
backend**



**Developer-friendly  
with React-based  
tools**



## Amplify React

A frontend development platform for building secure, scalable and cloud-connected web applications

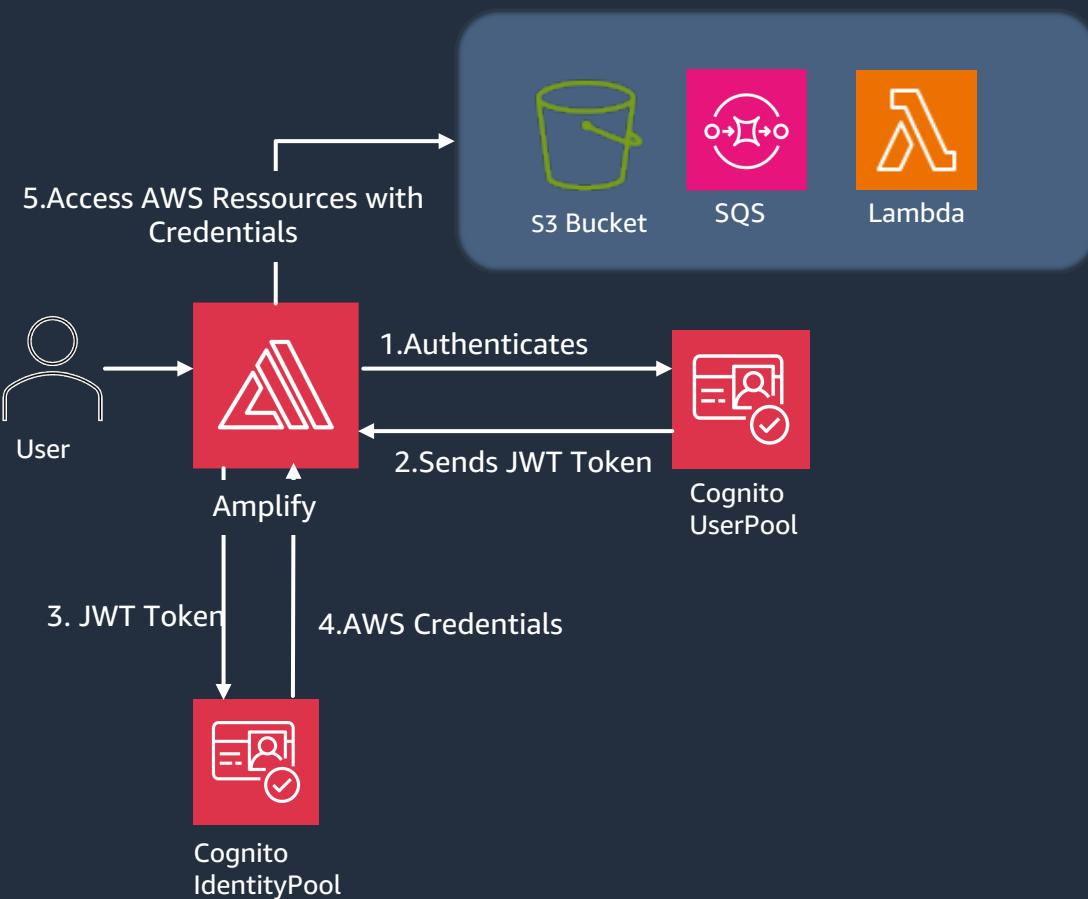
**Rapid deployment  
with pre-built UI  
components**



**Integrated with Cognito  
for secure  
authentication**



# Authentication and Authorization



## Amazon Cognito

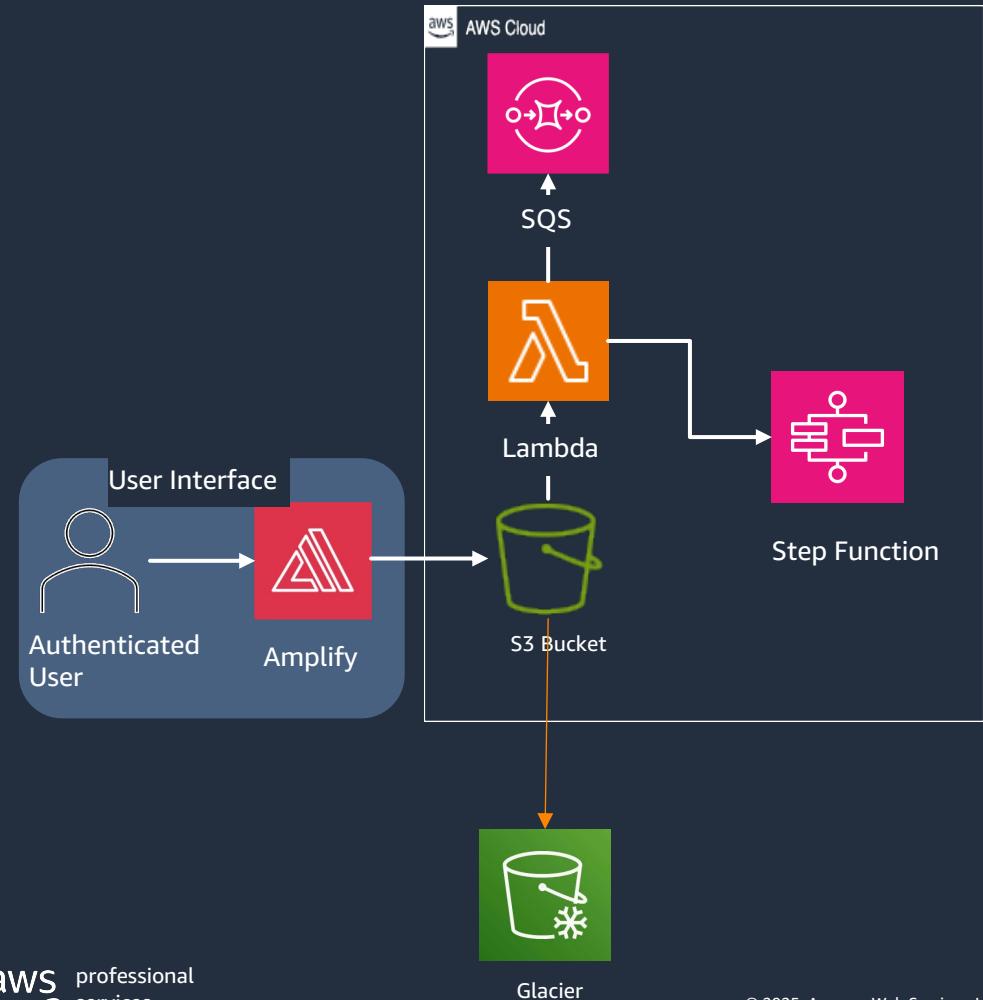
A fully managed AWS service that provides authentication, authorization, and user management for web and mobile applications.

- **User Pools**: handles authentication (WHO you are)
- **Identity Pools** : handles the authorization (WHAT you can access)

### Keys features:

- Zero credential storage in frontend.
- Temporary credentials
- No sign up, only sign in
- Role-based access control with least privilege principle:
  - Admin Group: can access the PII
  - Customer Group: can't access PII

# File Upload



## Keys features:

- Up to 400 documents at once
- Supported file types: .pdf, .png, .jpeg, .tiff
- 500 MB/ file
- Secure pre-signed URLs with 2-hour expiration for enhanced security
- S3 versioning
- Automatic transition to Glacier storage after 90 days
- SQS messages contain:
  - Workflow ID
  - Document name
  - Input path

# Live Demo...

Deliver Results

# Step Function workflow:

- Text Extraction
- PII/PHI Detection
- Redaction

# 1. Text Extraction – more than just OCR...

## Structured data extraction

Extracts txt, forms, tables, lines.  
From: pdf, JPEG,PNG



## Asynchronous processing



## Amazon Textract

AI service that automatically extracts text, handwriting, and data from scanned documents.

- Highly accurate
- High confidence score

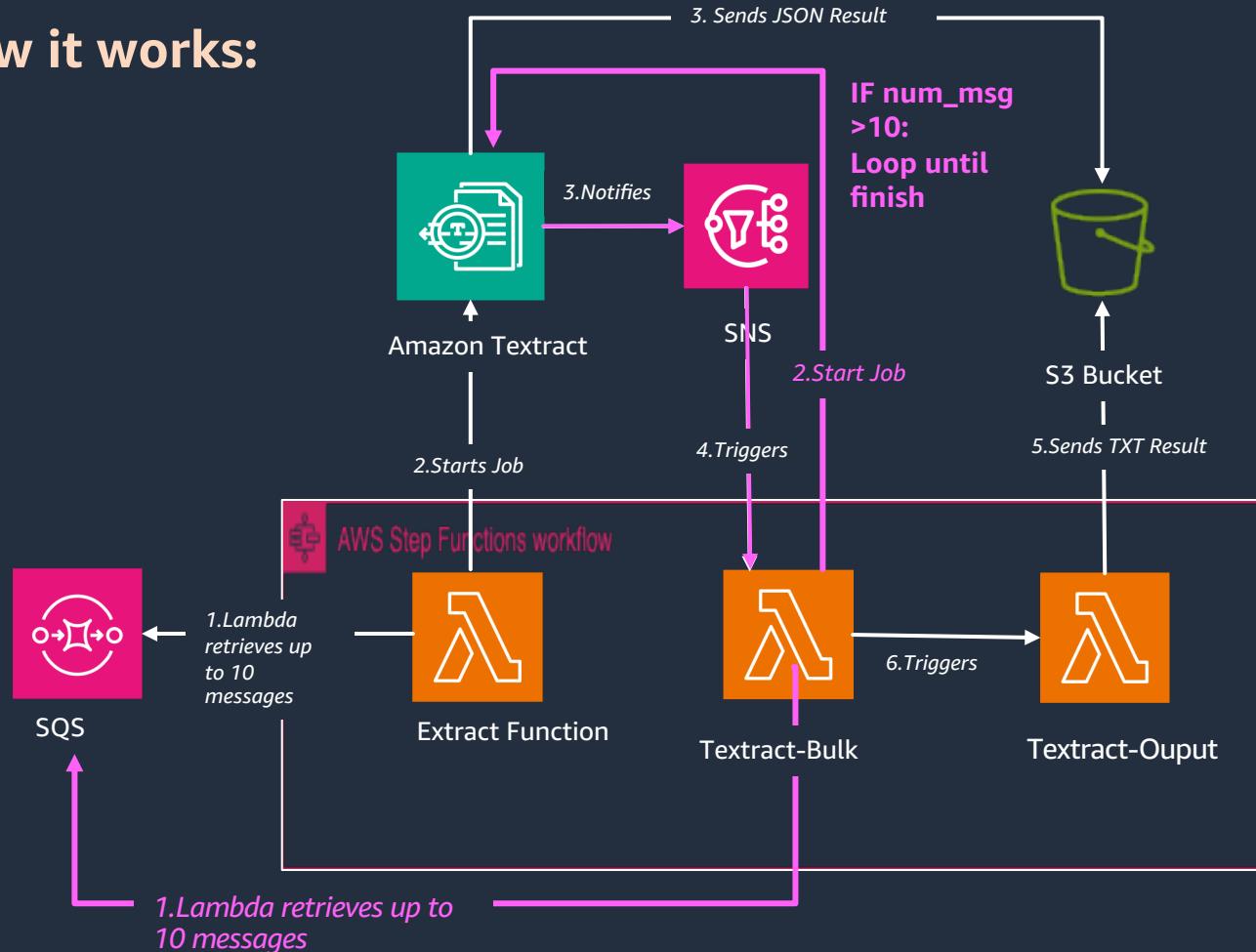


No ML training required



# 1. Text Extraction

**How it works:**



## Text extraction Workflow:

- Max SQS receive\_message=10 (Prevents overwhelming the system )
- Lambda retrieves up to 10 messages from SQS :
- Using SNS Eliminates resource-intensive polling

**Textract-bulk:** Orchestrates the workflow

- Triggers Textract-Output
- If num\_msg >10: Loop until all documents are processed
- **Textract-output:** Transforms raw Textract JSON to TXT

# Code Snippets..

Dive deep

# Code Snippets... Dive deep

```
txrct_response = textract.start_document_analysis(  
    DocumentLocation={  
        'S3Object': {  
            'Bucket': env_vars['INPUT_BKT'],  
            'Name': f"public/{doc['input_path']}{doc['document_name']}",  
        },  
        FeatureTypes=['TABLES', 'FORMS'],  
        JobTag=doc['workflow_id'],  
        NotificationChannel={  
            'SNSTopicArn': env_vars['SNS_TOPIC'],  
            'RoleArn': env_vars['SNS_ROLE']  
        },  
        OutputConfig={  
            'S3Bucket': env_vars['INPUT_BKT'],  
            'S3Prefix': f"public/output/{doc['workflow_id']}"  
        }  
)
```

Initiating Textract job

# Code Snippets... Dive deep

```
# Some documents remain to be submitted to Textract
if len(processed_files) < len(files):    You, 3 weeks ago * stacks and resources
    # Documents remained to be processed or are being processed
    event["bucket"] = bucket
    jobs = get_msg_submit(event, env_vars, 10)
    logger.debug(f"Submitted Jobs : {json.dumps(jobs)}")
    return jobs
else:
    #Post to state machine that workflow is done
    select = f"SELECT \"workflow_token\" FROM \"{env_vars['PII_TABLE']}\" WHERE part_key=? AND sort_key=?"
    ddb_response = ddb.execute_statement(Statement=select, Parameters=[
        {'S': workflow_id},
        {'S': f"input/{workflow_id}/"})
    )
    deserialized_document = {k: deserializer.deserialize(v) for k, v in ddb_response['Items'][0].items()}
    sm_token = deserialized_document['workflow_token']
    smresponse = sfn.send_task_success(taskToken=sm_token,
                                         output=json.dumps({
                                             "Payload": {
                                                 "workflow_id": workflow_id,
                                                 "bucket": bucket,
                                                 "tmp_process_dir": f"{root_prefix}/temp/{workflow_id}",
                                                 "phi_input_dir": f"{root_prefix}/phi-input/{workflow_id}"
                                             }
                                         }))
    logger.debug(smresponse)
```

Loop

## 2. PII/PHI Detection



## 2. PII Detection – Amazon Comprehend Medical

How it works:

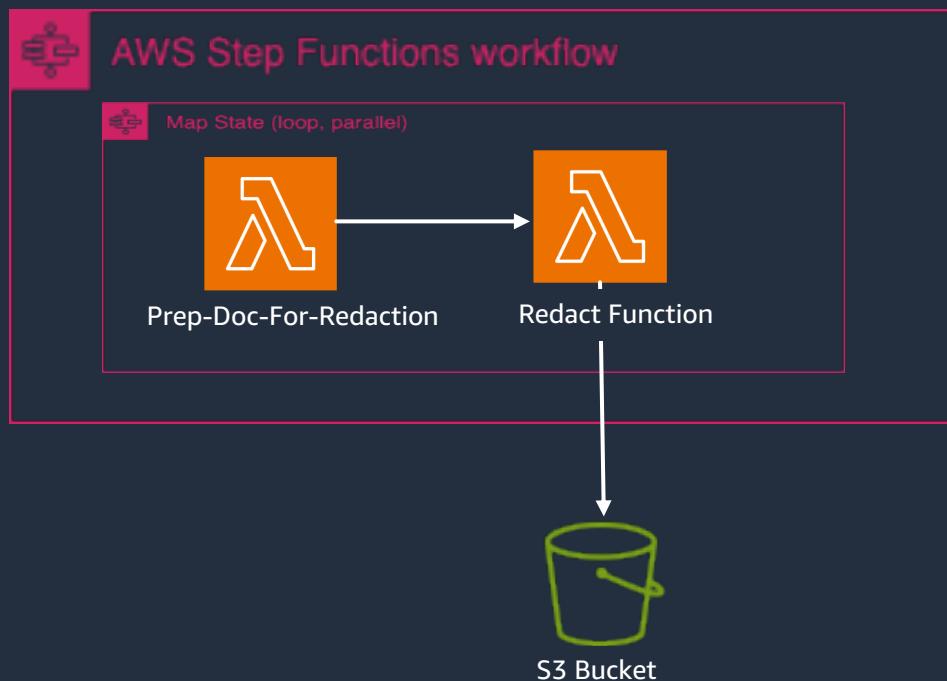


### Key Features:

- Automated Detection & Redaction
- asynchronous API
- Parallel processing of multiple documents
- HIPAA-compliant

# 3. PII Redaction - Lambda

## How it works:



### Map State:

- Maximum scale: 40 parallel executions × 10 documents each = 400 documents simultaneously

### 1. Prep-Doc-For-Redaction :

maps between:

- Original document location
- Textract output
- Amazon Comprehend Medical Output

### 2. Redaction: :

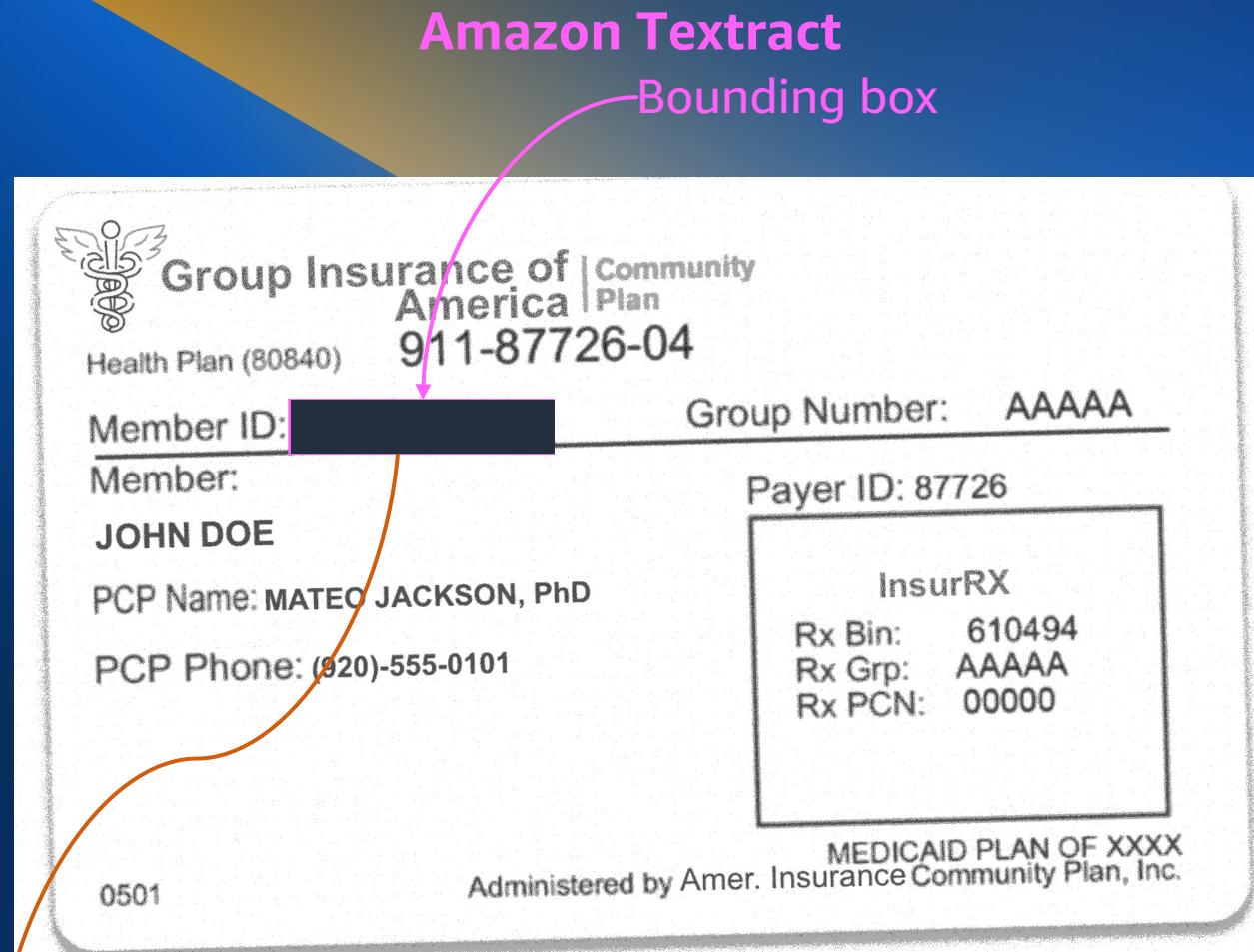
- Visual Redaction Application:

- Use Textract bounding boxes
- Draws black rectangles precisely over PHI entities (detected by comprehend medical) on origal files

- Map state updates workflow status in DynamoDB

# Redaction:

Redact.py lambda function



Amazon Comprehend Medical

# Code Snippets..

Dive deep

# Code Snippets... Dive deep

## 1. Getting the bounding boxes for all text lines in the document

```
# Set up overlay for text lines
logger.debug("Setting overlay")
overlay=[Textract.Types.LINE]

# Get bounding boxes for text from Textract JSON
logger.debug("Getting bounding boxes")
bounding_box_list = get_bounding_boxes(textract_json=textract_json, document_dimensions=document_dimension, overlay_features=overlay)
```

## 2. Getting the PHI entities from Comprehend Medical

```
for entity in comprehend_json['Entities']:
    entity_text = entity['Text']
    entities.append(entity_text)
    # Add lowercase entity to lookup dictionary
    entity_lookup[entity_text.lower()] = True
```

# Code Snippets... Dive deep

```
for idx, bbox in enumerate(bounding_box_list): _____For every bounding box
    if idx in processed_boxes:
        continue
    bbox_text_lower = bbox.text.lower()

    for entity in entities: _____For every PHI
        entity_lower = entity.lower()
        if entity_lower in bbox_text_lower or bbox_text_lower in entity_lower:
            redactions.append(bbox)
            processed_boxes.add(idx)
            break
```

matching text entities from Comprehend Medical with their exact physical locations from Textract.

# Code Snippets... Dive deep

```
# Draw black rectangles over bounding boxes that contain PHI entities
for idx,img in enumerate(images):
    draw = ImageDraw.Draw(img)
    page_num = idx + 1
    for box in redactions:
        if box.page_number == page_num:
            draw.rectangle(xy=[box.xmin, box.ymin, box xmax, box ymax], fill="Black")
```

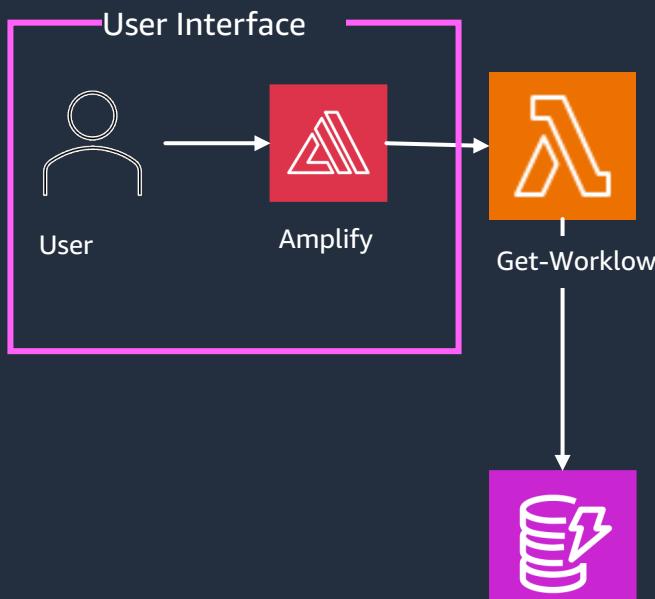
The final snippet applies the actual redactions by drawing black rectangles precisely over sensitive information.

# Observability and Monitoring

Insist on High Standards...

# FrontEnd – Status Tracking

## How it works:



Review Document Status

Workflow ID	Number of Docs	OCR Status	Redaction Status	Time Submitted
6271cb5b-72a9-4939-b5bc-f28a8a8cad7d	5	✓ Complete	✓ Complete	07/15/2025 11:32 AM
c8b44113-54bd-4cee-80db-52b40ccae08d	5	✓ Complete	: Processing	07/15/2025 11:09 AM
11147eb3-9c27-4c96-a48d-07867096320a	4	✓ Complete	: Processing	07/15/2025 10:52 AM

# Monitoring and audit – CloudWatch

## Log Groups

The screenshot shows the AWS CloudWatch Log Groups interface. The left sidebar includes sections for Dashboards, AI Operations (Overview, Investigations, Configuration), Alarms (In alarm, All alarms, Billing), Logs (Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights), Metrics, Application Signals (APM), and Network Monitoring. The main area displays a table titled "Log groups (28)" with columns for Log group, Log class, Anomaly detection, Data protection, Sensitive data, Retention, Metric filters, and Controls. Each row lists a specific log group path, such as "/aws/lambda/med-get-workflow". The interface includes a search bar, filter options (Exact match), and buttons for Actions, View in Logs Insights, Start tailing, and Create log group.

## Alarms

The screenshot shows the AWS CloudWatch Alarms interface. The left sidebar includes sections for Dashboards, AI Operations (Overview, Investigations, Configuration), Alarms (In alarm, All alarms, Billing), Logs (Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights), Metrics, Application Signals (APM), and Network Monitoring. The main area displays a table titled "Alarms (2)" with columns for Name, State, Last state update (UTC), Conditions, and Actions. Two alarms are listed: "TargetTracking-table/med-reduction-AlarmLow-cc1af3ca-030f-4a65-8a10-660caeae3ce19" and "TargetTracking-table/med-reduction-AlarmLow-35ace7e6-15a9-43a2-bf82-7b8ff8fabee07", both of which are currently in an alarm state. The interface includes a search bar, filter options (Hide Auto Scaling alarms, Clear selection, Create composite alarm, Actions), and a button for Create alarm.

# Monitoring – CloudTrail (API)

The screenshot shows the AWS CloudTrail Event history interface. The left sidebar includes links for Dashboard, Event history (which is selected), Insights, Lake (with sub-links for Dashboards, Query, Event data stores, and Integrations), Trails, Settings, Pricing, Documentation, Forums, and FAQs. The main content area has a blue header bar with a message about enriching events with resource tags and IAM global keys. Below this is a section titled "Event history (50+)" with a "Info" link. It states that the history shows the last 90 days of management events. A "Lookup attributes" table is displayed, showing columns for Event name, Event time, User name, Event source, Resource type, and Resource name. The table lists ten events, all of which are "CreateLogStream" events from July 15, 2025, at various times between 16:04:03 and 16:50:43 UTC+0. The "Event name" column contains hyperlinks to the event details. The bottom of the table shows "0 / 5 events selected".

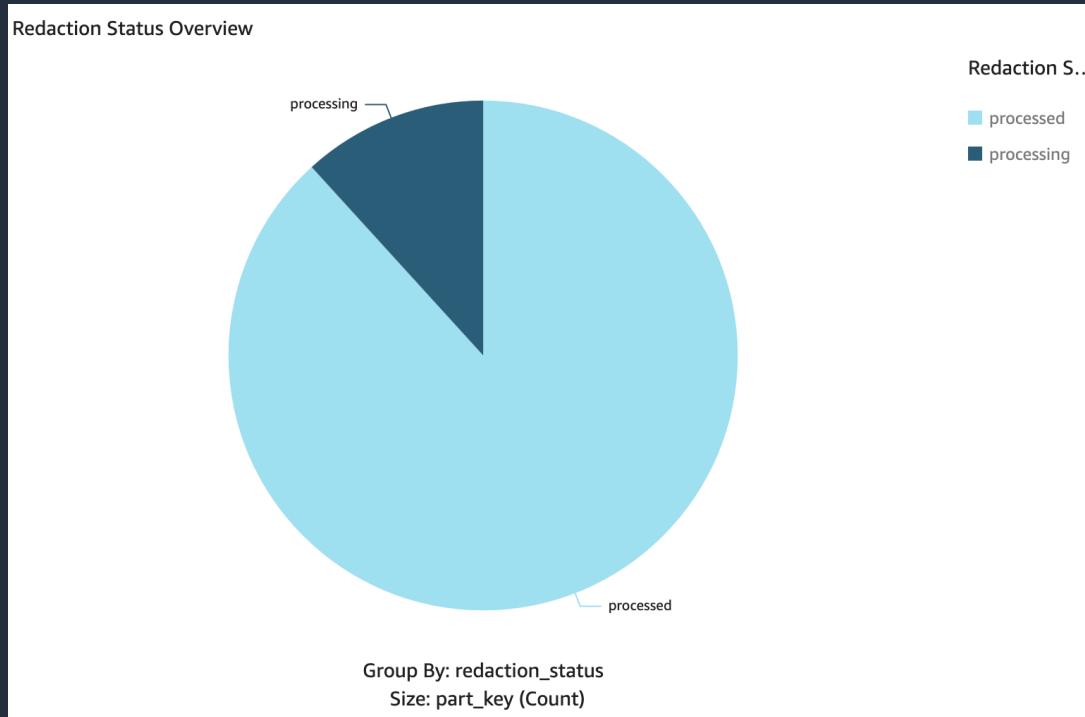
Event name	Event time	User name	Event source	Resource type	Resource name
CreateLogStream	July 15, 2025, 16:50:43 (UTC+0...)	med-get-workflow	logs.amazonaws.com	-	-
CreateLogStream	July 15, 2025, 16:34:58 (UTC+0...)	med-redact-docu...	logs.amazonaws.com	-	-
CreateLogStream	July 15, 2025, 16:27:34 (UTC+0...)	med-get-workflow	logs.amazonaws.com	-	-
InitiateAuth	July 15, 2025, 16:27:23 (UTC+0...)	-	cognito-idp.amazonaws.com	-	-
RespondToAuthChall...	July 15, 2025, 16:27:23 (UTC+0...)	-	cognito-idp.amazonaws.com	-	-
RevokeToken	July 15, 2025, 16:18:37 (UTC+0...)	-	cognito-idp.amazonaws.com	-	-
CreateLogStream	July 15, 2025, 16:15:53 (UTC+0...)	med-get-workflow	logs.amazonaws.com	-	-
InitiateAuth	July 15, 2025, 16:15:50 (UTC+0...)	-	cognito-idp.amazonaws.com	-	-
CreateLogStream	July 15, 2025, 16:10:24 (UTC+0...)	med-redact-docu...	logs.amazonaws.com	-	-
CreateLogStream	July 15, 2025, 16:04:03 (UTC+0...)	med-redact-docu...	logs.amazonaws.com	-	-

# BI (Business Intelligence) - QuickSight

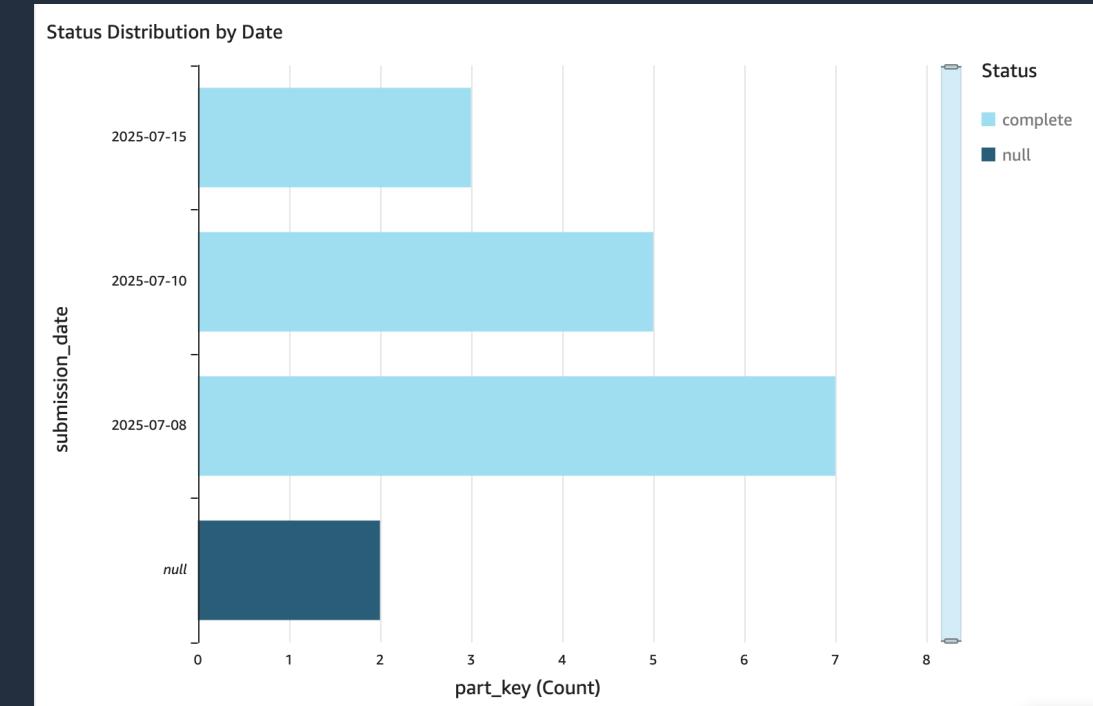
High Level:



# QuickSight Dashboard – Pie Chart

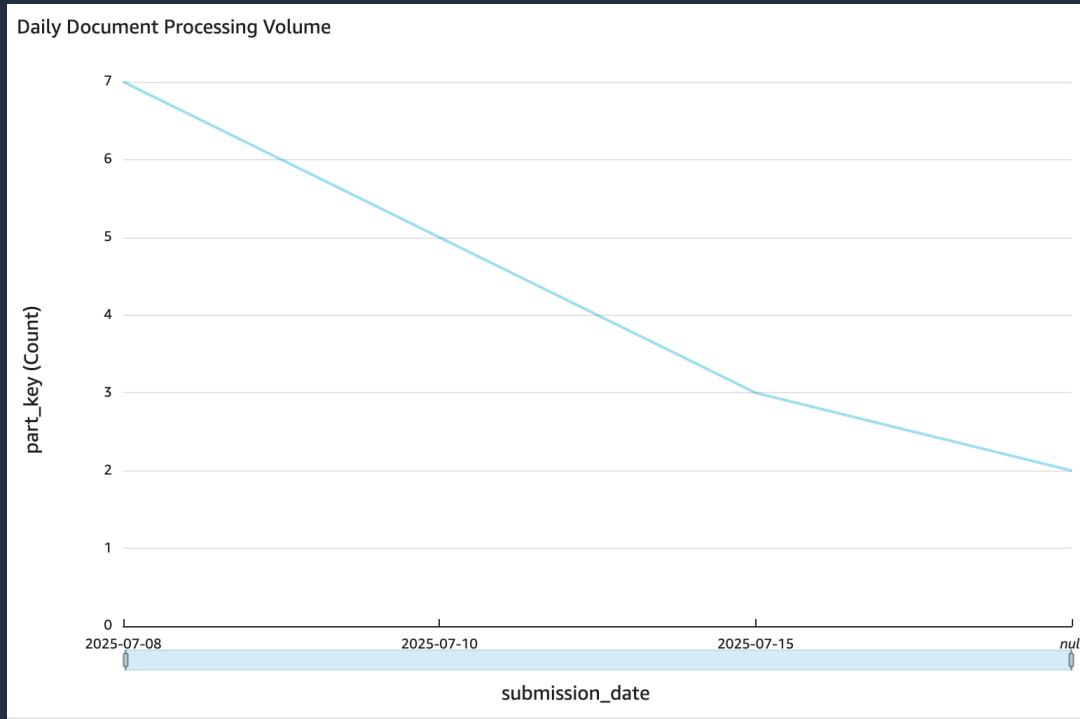


# Pie Chart

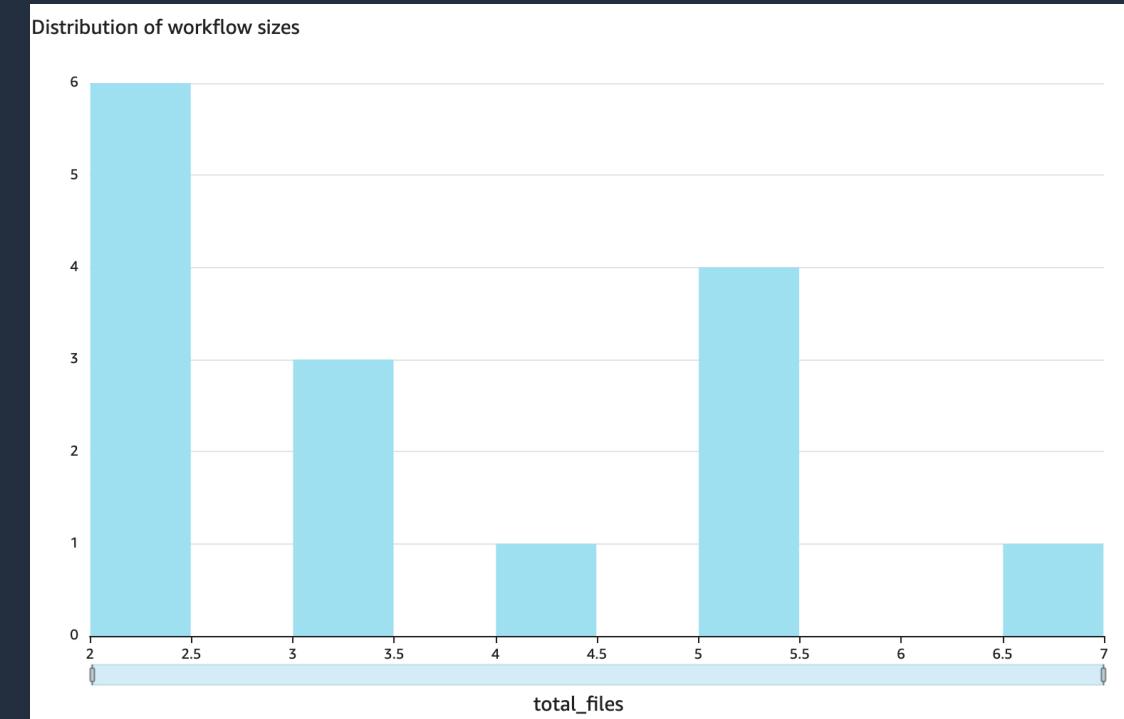


## Stacked bar chart

# QuickSight Dashboard



Line Chart



histogram

# Security

**Insist on High Standards...**

**Amazon's TOP Priority !**

# Security



Secure storage  
with encryption  
at rest / in transit



IAM roles



KMS Keys



Permissions  
Least Privilege Principle



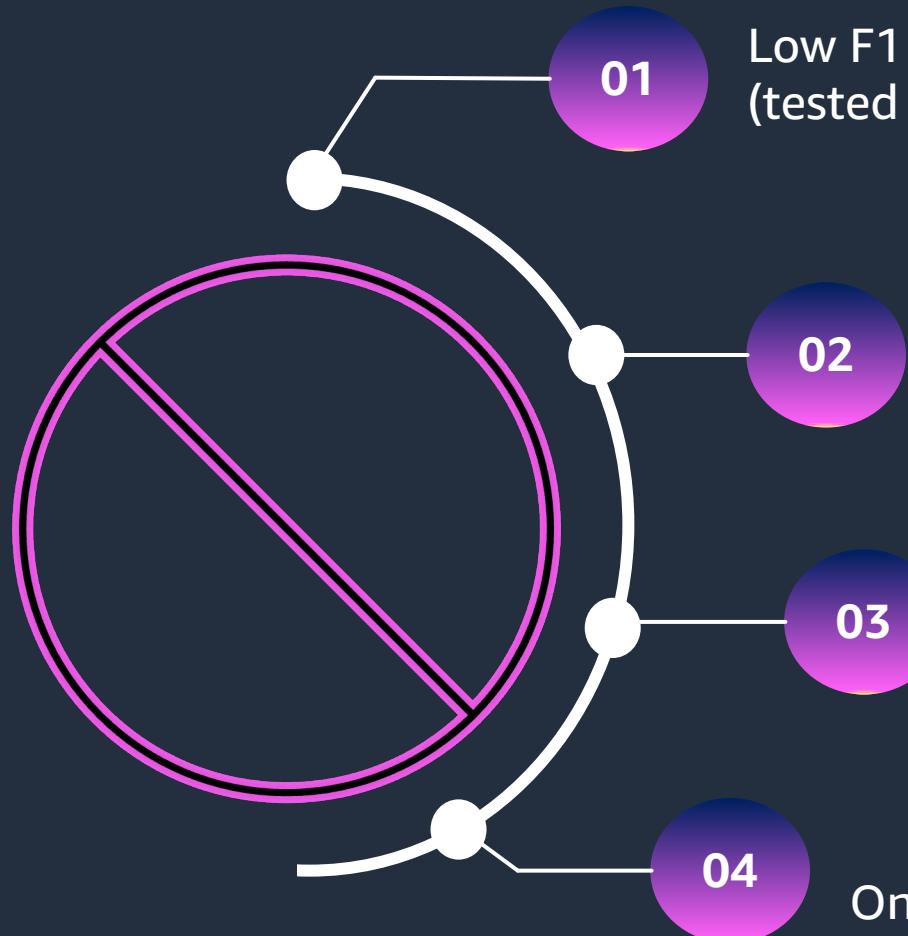
RBAC

# Back to the Live Demo...

Deliver Results

# Limitations

# Limitations

- 
- 01 Low F1 score Amazon Comprehend Medical of 0.77 (tested with 20 documents)
  - 02 English-Language Documents
  - 03 Batch Processing takes 8 minutes
  - 04 Only accepts PDF, PNG, JPEG, TIFF document types

# Next Step

# Think Big



## Next Steps

- Multi-region
- CloudFront
- Route 53 domain name
- Multi industries not just Healthcare
- Extend support for other format types (conversion)
- Implement custom entity recognizers using Amazon SageMaker to supplement Comprehend Medical
- Integrate Amazon Translate as a pre-processing step for non-English documents

# Lessons Learned



## Challenges

- Thinking Big
- Bias for Action
- Insist on high standards
- Working through Ambiguity



## Victories

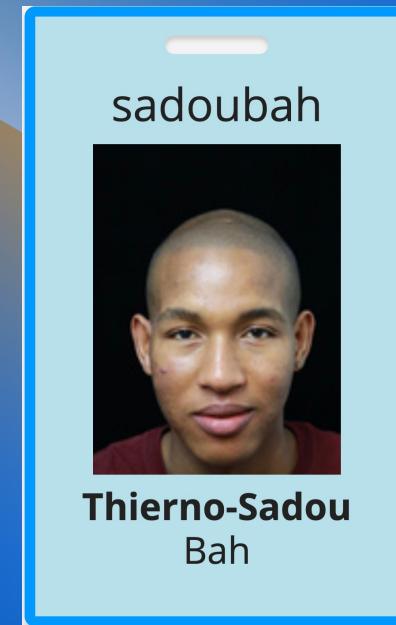
- Taking Ownership
- Learning and being curious
- Diving deeper
- Customer Obsession



## Takeaways

- Networking
- Earning trust
- Discipline

# Acknowledgements





# Thank you!

You are now welcome to ask any questions.

Miniar Jabri

@miniarja