Implementation Guide

Enhanced Document Understanding on AWS

Enhanced Document Understanding on AWS: Implementation Guide

Table of Contents

Deploy an event-driven AWS Solution that automates document ingestion, analysis, detection, and redaction

Organizations across industries are increasingly required to process large volumes of semi-structured and unstructured documents with greater accuracy and speed. They need a document processing system that ingests and analyzes documents, extracts their content, identifies and redacts sensitive customer information, and creates search indexes from the analyzed data.

Many industries have stringent compliance requirements to redact personally identifiable information (PII) and protected health information (PHI) from documents. In most cases, organizations manually process documents to extract information and insights. This approach can be time consuming, expensive, and difficult to scale. Organizations need information to rapidly extract insights from documents. They can benefit from a smart document processing system as a foundation to automating business processes that rely on manual inputs and interventions.

To help meet these needs, the Enhanced Document Understanding on AWS solution:

- Automates document ingestion process to improve operational efficiency and reduce cost.
- Ingests and analyzes document files at scale using artificial intelligence (AI) and machine learning (ML).
- Extracts text from documents.
- Identifies structural data (such as single word, a line, a table, or individual cells within a table).
- Extracts critical information (such as entities).
- Creates smart search indexes from the data.
- Detects and redacts PII and PHI to generate a redacted version of the original document.

You can use each of these features standalone or configure the solution as a unique composition of workflow orchestration based on your use case.

The solution also provides a web user interface (UI) for users to upload documents. Once the documents are uploaded, a backend workflow orchestrates AWS managed AI services to process documents at scale.

This implementation guide provides an overview of the Enhanced Document Understanding on AWS solution, its reference architecture and components, considerations for planning the

deployment, and configuration steps for deploying Enhanced Document Understanding on AWS to the Amazon Web Services (AWS) Cloud.

The intended audience for implementing this solution in their environment includes solution architects, business decision makers, DevOps engineers, data scientists, and cloud professionals.

Use this navigation table to quickly find answers to these questions:

If you want to . . .	Read . . .
Know the cost for running this solution. The estimated cost for running this solution in the US East (N. Virginia) Region is USD $1,847.28 per month.	Cost
Understand the security considerations for this solution.	Security
Know how to plan for quotas for this solution.	Quotas
Know which AWS Regions are supported for this solution.	Supported AWS Regions
Know how to configure different workflow options to meet business needs	Architecture details
View or download the AWS CloudForm ation template included in this solution to automatically deploy the infrastructure resources (the “stack”) for this solution.	AWS CloudFormation template
Access the source code and optionally use the AWS Cloud Development Kit (AWS CDK) to deploy the solution.	GitHub repository

Features and benefits

This solution provides the following features:

Extensible modular architecture

We architected this solution an event-driven architecture paradigm. This approach provides the flexibility to extend the existing features by adding new workflow components (for example, custom classification and detection of domain specific entities). The custom Amazon EventBridge event bus forms the central hub for events and configuring consumers or listeners.

We modularly structured the AWS CloudFormation template as a root template with nested templates. Each nested template creates resources that are specifically required by the relevant function. Based on the features required for a use case, you can configure the root template to deploy some or all of the nested templates.

Customizable workflow orchestration

You can customize workflow orchestration to align with your unique use case. To help you understand the configuration and provide initial scaffolding, the application has out-of-the-box workflow configuration definitions to choose from. The choice of workflow configuration drives the nested templates to be deployed through CloudFormation.

AI and ML to aid processing documents at scale

The solution uses AWS managed AI services Amazon Textract, Amazon Comprehend, and Amazon Comprehend Medical. This solution allows businesses with little or no knowledge of or training in deploying ML models to start automating their document processing.

Custom Amazon CloudWatch dashboard and custom metrics

To provide observability, the solution publishes custom metrics information. The solution creates a custom Amazon CloudWatch dashboard to chart these metrics along with the default metrics provided by CloudWatch.

Integration with Service Catalog AppRegistry and Application Manager, a capability of AWS Systems Manager

This solution includes a Service Catalog AppRegistry resource to register the solution's CloudFormation template and its underlying resources as an application in both AppRegistry and Application Manager. With this integration, you can centrally manage the solution's resources.

Use cases

Account administration

To support opening accounts and processing account documents such as application forms and claims, this solution can automate extracting content from:

- Application forms (as key-value pairs)
- ID documents (such as a driver's license or passport)
- Paystubs
- Utility bills

Since we built this solution with an event-driven architecture, you can:

- Integrate the inference outputs from the solution with an enterprise's internal systems by invoking internal application programming interfaces (APIs).
- Use the PII detection and redaction feature of the solution to redact PII content in documents.
- Extend the current implementation to restrict unredacted documents to a selected group of users.
- Configure extracted content from these processes to feed into your internal systems, leading to further processing and expediting services.

Handling PII and PHI

Under the Health Insurance Portability and Accountability Act (HIPAA), PHI that is a part of 18 identifiers must be treated with special care and only be accessed by authorized personnel. Organizations that handle PII and PHI, such as healthcare and life sciences organizations, need to be cognizant of PII and PHI data. You can configure this solution's workflows to detect PHI by using Amazon Comprehend Medical. You can also perform redaction on the processed documents. By extending the default solution implementation, you can restrict the unredacted documents to a select user-group authorized to view PHI under Safe Harbor guidelines.

Life sciences and research organizations, for example, can optimize the matching process for enrolling patients into clinical trials. By using Amazon Comprehend Medical to detect pertinent information in clinical text, researchers can improve pharmacovigilance, perform post-market surveillance to monitor adverse drug events, and assess therapeutic effectiveness by detecting vital information in follow-up notes and other clinical texts.

Customers in the healthcare insurance sector, for another example, can expand their analytics to include unstructured documents such as clinical notes. You can analyze detailed information about medical diagnoses to help determine appropriate billing codes from unstructured documents.

Natural language processing (NLP) is the most critical component of computer-assisted coding (CAC). You can use Amazon Comprehend Medical Named Entity and Relationship Extraction (NERe) APIs to analyze clinical text, helping to decrease time to revenue and improve reimbursement accuracy.

Maintenance logs

Organizations that keep maintenance logs of heavy machinery and equipment, such as manufacturing and energy organizations, can benefit from automated document processing. Maintenance logs are often handwritten and need to be digitized. This solution uses Amazon Textract to extract information from handwritten documents to automate digitizing record keeping.

Concepts and definitions

This section describes key concepts and defines terminology specific to this solution:

case

A logical group of related documents that the solution processes together.

inference

Refers to the result of a workflow performed on a document, such as the text extracted using Amazon Textract in the text extraction workflow. The solution stores inferences in Amazon Simple Storage Service (Amazon S3). You can retrieve them using the `/inferences/` endpoint of the REST API provided by this solution. See the API reference for further details.

stage

The running of a single workflow performed on documents in a case as part of a sequence of workflows as defined in the workflow configuration.

workflow

A processing operation that you can perform on documents in a case, such as text extraction and entity detection. See the Architecture details for descriptions of the available workflows included in this solution

workflow configuration

Defined in a JSON file, a workflow configuration specifies an ordered set of workflows (or stages), and a set of required documents and conditions (including file types and size limits). When you deploy the solution, you must provide a workflow configuration to determine how the solution processes cases. See the Customization guide for details on workflow configurations.

workflow orchestrator

This AWS Lambda function is at the core of the internal orchestration logic of this solution. The workflow orchestrator listens to several types of EventBridge events and runs the correct workflow AWS Step Functions in order for a case depending on these input events.

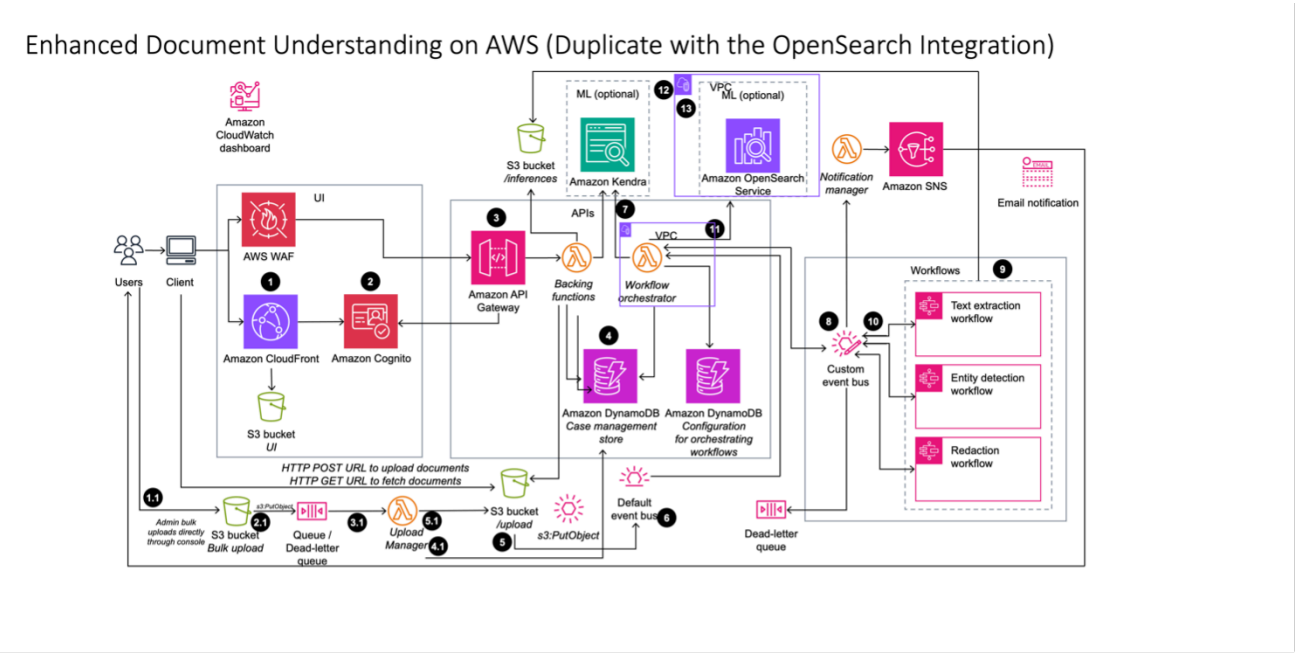
> **Note**
>
> For a general reference of AWS terms, see the AWS Glossary.

Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this solution.

Architecture diagram

Deploying this solution with the default parameters deploys the following components in your AWS account.



Enhanced Document Understanding on AWS architecture

> **Note**
>
> CloudFormation resources are created from AWS Cloud Development Kit (AWS CDK) constructs.

The high-level process flow for the solution components deployed with the CloudFormation template is as follows:

1. The user requests the browser to navigate to an Amazon CloudFront URL.
2. The UI prompts the user for authentication, which the solution validates using Amazon Cognito.

3. The UI interacts with the REST endpoint deployed on Amazon API Gateway.
4. The user creates a case that the solution stores in the **Case management store** Amazon DynamoDB table.
5. The user requests a signed Amazon Simple Storage Service (Amazon S3) URL to upload documents to an S3 bucket.
6. Amazon S3 generates an `s3:PutObject` event on the default Amazon EventBridge event bus.
7. The `s3:PutObject` event invokes the workflow orchestrator AWS Lambda function. This function uses the configuration stored in the **Configuration for orchestrating workflows** DynamoDB table to determine the workflows to be called.
8. The workflow orchestrator Lambda function creates an event and sends it to the custom event bus.
9. The custom event bus invokes one of the three AWS Step Functions state machine workflows based on the event definition.
10. The workflow completes and publishes an event to the custom EventBridge event bus.
11. The custom EventBridge event bus invokes the workflow orchestrator Lambda function. This function uses the configuration stored in the **Configuration for orchestrating workflows** DynamoDB table to determine whether the sequence is complete or if the sequence requires another workflow:
    a. The solution updates the **Case management store** DynamoDB table.
    b. If the sequence is not complete, the solution returns to step 8 for the next state machine workflow.
12. (Optional) The workflow orchestrator Lambda function writes metadata from the processed information to an Amazon Kendra index. This index provides the ability to perform ML powered search.

> **Note**
>
> The deployment to Amazon Kendra is optional. If not deployed the search feature is not available.

13. (Optional) The workflow orchestrator Lambda function writes metadata from the processed information to an Amazon OpenSearch Serverless collection. This collection provides the ability to perform keyword search.
14. (Optional) Open Search is powered by AWS OpenSearch Serverless and the OpenSearch collection is protected by running in Vpc – 2 private subnets. The Vpc currently provisions

a security group that allows all outbound traffic from OpenSearch and an ingress rule for Lambda to write inferences. The Vpc also provisions 2 interface endpoint (AWS PrivateLink) that allows both Lambda and KMS to access the Open Search collection. KMS does not directly access OpenSearch but it is used for storing and managing the encryption keys to perform the encryption of data at rest.

> **Note**
>
> The deployment to Amazon OpenSearch Serverless is optional. If not deployed the search feature is not available.

AWS Well-Architected design considerations

This solution was designed with best practices from the AWS Well-Architected Framework which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how the design principles and best practices of the Well-Architected Framework were applied when building this solution.

Operational excellence

This section describes how we architected this solution using the principles and best practices of the operational excellence pillar.

- We built the solution as infrastructure as code using CloudFormation.
- Lambda functions push custom metrics to CloudWatch and a custom CloudWatch dashboard to monitor the health of the solution.
- The solution components are modularized, providing the flexibility to choose which components to deploy.

Security

This section describes how we architected this solution using the principles and best practices of the security pillar.

- The solution encrypts data at-rest and in-transit.

- All service resources integrate through AWS Identity and Access Management (IAM) roles using the best practice of least-privilege permissions.
- Policy definitions don't use AWS managed policies.
- Each Lambda function has its own role and policy (no shared roles between Lambda functions).
- CloudFront and AWS WAF protect publicly-exposed endpoints.

Reliability

This section describes how we architected this solution using the principles and best practices of the reliability pillar.

- We built the solution to use a serverless architecture wherever possible.
- We built the architecture for on-demand, horizontal scalability, and automatic recovery from underlying infrastructure failure.
- The architecture includes buffering and throttling requests to not overwhelm underlying endpoints.
- We orchestrated the workflows AWS Step Functions to state management and retry failures.
- We configured the dead-letter queue to capture failures and retry failed requests.

Performance efficiency

This section describes how we architected this solution using the principles and best practices of the performance efficiency pillar.

- The architecture uses DynamoDB, a fully managed serverless NoSQL database with on-demand scaling.
- The architecture uses Amazon S3 as object storage and to host a website (through CloudFront) to provide low cost and scalability, with 99.999999999% durability.

Cost optimization

This section describes how we architected this solution using the principles and best practices of the cost optimization pillar.

- We built the solution with a serverless architecture, and customers pay only for what they use.

- The architecture uses pre-trained models and endpoints from AWS AI services. No dedicated provisioned endpoints for machine learning inferences.

Sustainability

This section describes how we architected this solution using the principles and best practices of the sustainability pillar.

- The solution's modular, componentized architecture provides the flexibility to customize resources to provision for individual use cases.
- The architecture uses serverless compute and storage, which optimizes resource utilization.
- As a cloud-based solution, this solution benefits from shared resources, networking, power cooling, and physical facilities.

Architecture details

This section describes the components and AWS services that make up this solution and the architecture details on how these components work together.

This solution includes three separate Step Functions workflows invoked by EventBridge events, as described in the Architecture diagram section. The following sections describe each of these workflows, AWS services used in this solution, and how the solution works.

AWS services in this solution

AWS service	Description
Amazon API Gateway	**Core.** This service provides the REST API for the solution.
Amazon Cognito	**Core.** This service handles user management and authentication for the API.
Amazon Comprehend	**Core.** This service performs standard and PII entity detection. **Note** The CloudFormation template doesn't deploy this service. Instead, Lambda calls this service as a part of the architecture.
Amazon Comprehend Medical	**Core.** This service performs medical entity and PHI detection. **Note** The CloudFormation template doesn't deploy this service. Instead, Lambda

AWS service	Description
	calls this service as a part of the architecture.
Amazon DynamoDB	**Core.** Two tables contain data for this solution: • One table manages the state of the cases and documents processed by this solution. • One table contains workflow configurations.
Amazon EventBridge	**Core.** This solution uses an entirely event-dri ven architecture. • The default event bus listens for S3 upload events to the RequestProcessorDo cumentRepo, which invoke the workflow orchestrator Lambda function. • The solution uses the custom event bus for all other events related to workflow orchestration. This includes events to invoke and get responses from the workflow Step Functions, and sending success and failure notifications.
AWS KMS	**Core.** AWS managed keys provide server-side encryption on the Amazon SQS queues in this solution. The solution also uses AWS KMS to: • Encrypt communication with Amazon Textract • Manage keys for access to the deployed Amazon Kendra index (if deployed).

AWS service	Description
AWS Lambda	**Core.** The solution uses Lambda functions to: • Back the REST API endpoints • Handle the core logic of each workflow and the workflow orchestrator. • Implement custom resources during CloudFormation deployment for actions such as copying static files to Amazon S3 and populating the configuration database in DynamoDB.
Amazon SNS	**Core.** The solution creates an Amazon SNS topic to handle sending success and failure notifications to users through email.
Amazon SQS	**Core.** Amazon SQS acts as the intermediary between the workflow Step Functions and their core processing logic, which the solution implements as Lambda Functions. As such, the solution creates an Amazon SQS queue for each workflow. The solution also implement s a dead-letter queue with Amazon SQS to handle failed processing requests for each workflow.

AWS service	Description
Amazon S3	**Core.** This solution creates the following S3 buckets for storage: • `RequestProcessorDocumentRepo` – Stores documents uploaded by users of the UI or REST API • `RequestProcessorInferences` – Stores inferences from processing cases • `SetupAppConfig` - Stores email templates and acts as a staging bucket for workflow configuration files before they're loaded into DynamoDB. • `AccessLog` - Stores access logs for the other buckets in the solution.
AWS Step Functions	**Core.** Step Functions implement the workflows and interact with the workflow orchestrator with EventBridge events. Each workflow Step Function: • Handles the control flow to determine which documents in a case to process • Determines which Lambda function to use for processing each document • Sets input parameters to the Lambda functions • Implement retry mechanisms • Handle sending success and failure events to be picked up by the Workflow Orchestrator

AWS service	Description
Amazon Textract	**Core.** The solution uses Amazon Textract in the text extraction workflow to: • Perform OCR to extract raw text from uploaded documents in PDF or image form. • Extract key-value pairs and tabular content. **Note** The CloudFormation template doesn't deploy this service. Instead, Lambda calls this service as a part of the architecture.
AWS WAF	**Core.** The solution deploys a web applicati on firewall (WAF) in front of the API Gateway deployment to protect it.
AWS CloudFormation	**Supporting.** This solution is distributed as a CloudFormation template, and CloudForm ation deploys the AWS resources for the solution.
Amazon CloudWatch	**Supporting.** This solution publishes logs from solution resources to CloudWatch Logs, and publishes metrics for processed documents to CloudWatch metrics. The solutions also creates a CloudWatch dashboard to view this data, as well as CloudWatch Alarms to raise alerts when executions are failing.
AWS CDK	**Supporting.** The source code for this solution uses AWS CDK to generate the CloudForm ation templates.

AWS service	Description
IAM	**Supporting.** IAM manages access permissions between the resources in this solution, such as allowing a workflow Lambda function to write to the ML inferences S3 bucket. See IAM roles for details.
Service Catalog	**Supporting.** This solution uses Service Catalog AppRegistry to track and manage solution deployments.
Amazon Kendra	**Optional.** You can optionally deploy this solution with Amazon Kendra support, which provides NLP-based search for the uploaded documents.
Amazon OpenSearch	**Optional.** You can optionally deploy this solution with Amazon OpenSearch Serverles s support, which provides keyword search for the uploaded documents.

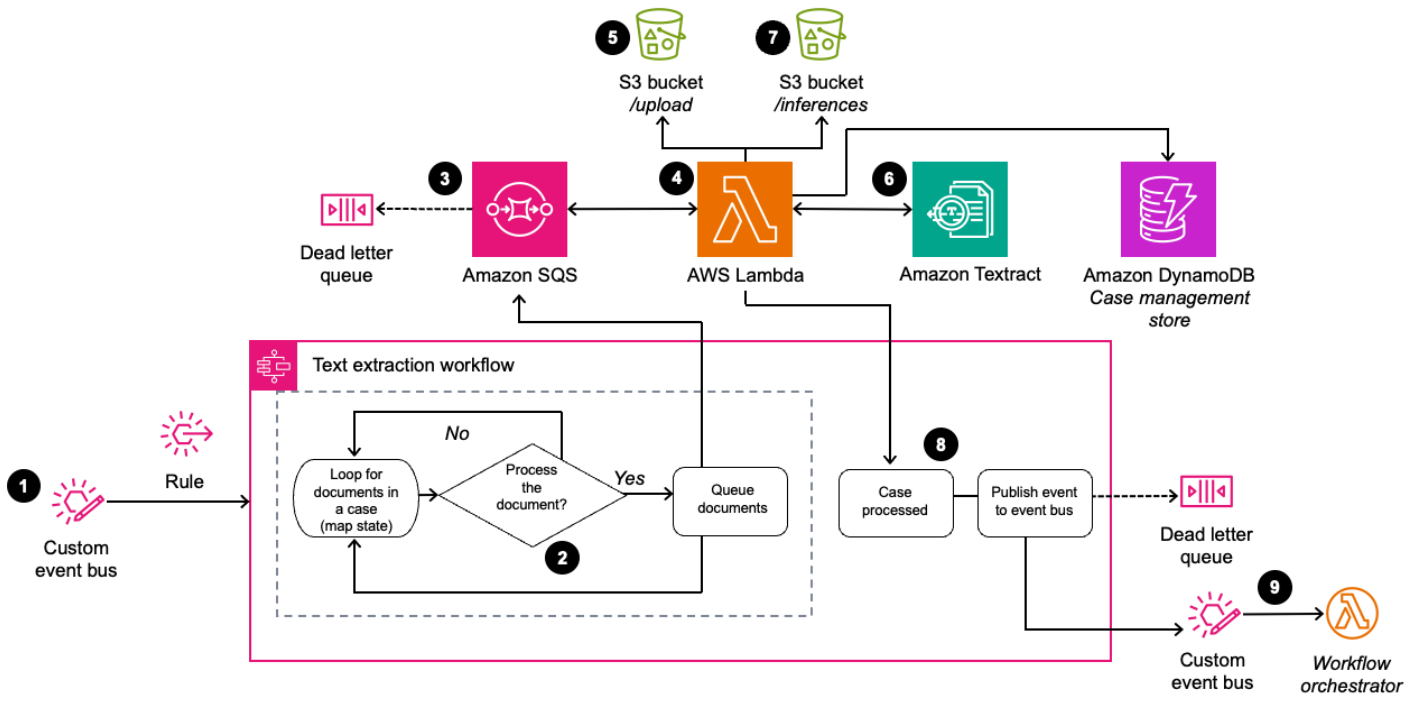
Text extraction workflow

The text extraction workflow extracts text from uploaded documents (images or `.pdf` files) using Amazon Textract.

Text extraction serves as the basis for:

- The entity detection workflow, to both perform the entity detection and map the entities to physical locations on the page.
- The redaction workflow, which depends on the entity locations to redact entities on the page.

> **Important**
>
> The text extraction workflow is required for all use cases, and you must run it before the entity detection workflow or redaction workflow.



Text extraction workflow

The process flow for the text extraction workflow is as follows:

1. An EventBridge custom event bus invokes a Step Functions state machine.
2. Based on the content of the event, the state machine determines whether the workflow should process each document.
3. An Amazon Simple Queue Service (Amazon SQS) queue pushes a message with metadata information for eligible documents (for example, the document location in Amazon S3 or the AWS API to use for analysis).
4. A Lambda function consumes the messages from the Amazon SQS queue.
5. The Lambda function retrieves the original document from the Documents S3 bucket, using the metadata information in the queue's message. If the document is a multi-page `.pdf` file,

the solution splits it into individual files for each page, then saves those files in the S3 bucket alongside the original document.

6. For each page, the solution calls Amazon Textract with one or more APIs:
   a. The DetectDocumentText API runs for every page. This API performs optical character recognition (OCR) on the provided document page and returns all text and their corresponding locations in the document.
   b. If you set the **RunAmazon TextractAnalyzeAction** parameter to `true` in the configuration file, then the solution runs an analysis action based on the `DocumentType` property of the current document. This analysis action is either AnalyzeDocument, AnalyzeID, or AnalyzeExpense. These analysis actions provide more domain-specific information about the extracted text of the documents. See How the solution works for more information.
7. The Lambda function uploads the results from all Amazon Textract API calls for all document pages to the ML inferences S3 bucket.
8. The Lambda function notifies the calling Step Function of success or failure.
   a. If the text extraction succeeds, this step is complete.
   b. If the text extraction fails, the solution publishes the event to an Amazon SQS dead-letter queue, configured with a default retention period of four days.
9. The Step Functions state machine publishes the success or failure event to the custom event bus, which invokes the workflow orchestrator Lambda function to create the next workflow event.

Entity detection workflow

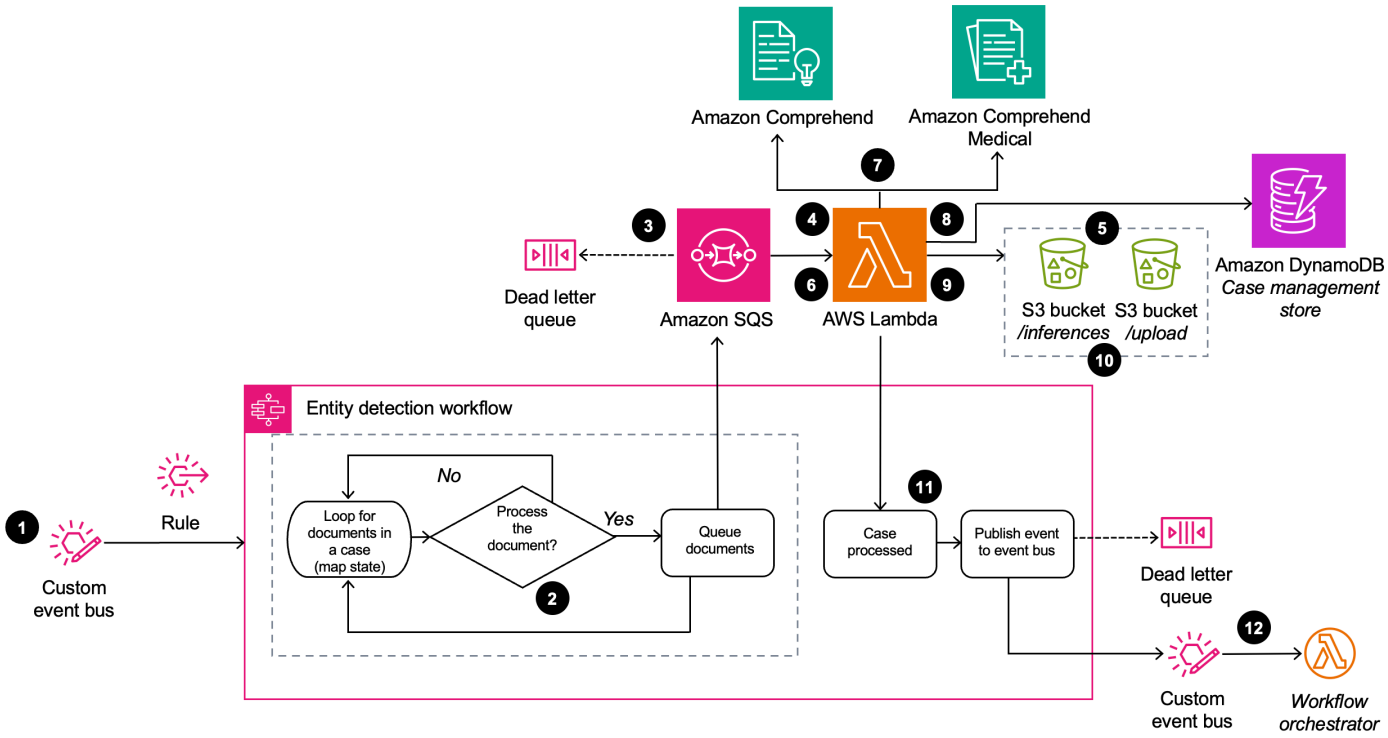
The entity detection workflow detects entities such as PII and medical entities in the extracted text from uploaded documents. This workflow uses Amazon Comprehend or Amazon Comprehend Medical depending on the configuration.

> **Important**
>
> You must run the text extraction workflow before the entity detection workflow.

This workflow returns entities that Amazon Comprehend or Amazon Comprehend Medical detect by their character offset. To map these entities to locations on a page, and thereby be able to display bounding boxes around text on the document, the workflow reconciles these results with

the Amazon Textract results from the text extraction workflow. The workflow then creates an inference, which maps entities to their physical locations (shown as bounding boxes in the UI) in the document.

> **Important**
>
> You must run the entity detection workflow before running the redaction workflow.



Entity detection workflow

The process flow for the entity detection workflow is as follows:

1. An EventBridge custom event bus invokes a Step Functions state machine.
2. Based on the content of the event, the state machine determines whether the workflow should process each document.
3. An Amazon SQS queue pushes a message with metadata information for eligible documents (for example, the document location in Amazon S3 or the AWS API to use for analysis).
4. A Lambda function consumes the messages from the Amazon SQS queue.

5. The Lambda function retrieves the inference containing the results of the Amazon Textract `DetectDocumentText` action created by the text extraction workflow from the ML inferences S3 bucket.
6. The Lambda function combines the text of each page of the document into a single string.
7. The Lambda function sends the strings representing the text on each page to Amazon Comprehend or Amazon Comprehend Medical. the service and API used are determined by the current config)
8. The Lambda function reconciles the Amazon Comprehend or Amazon Comprehend Medical results with the Amazon Textract results.
9. The Lambda function creates an inference that maps entities to their physical locations (shown as bounding boxes in the UI) in the document.
10. The Lambda function serializes both the raw Amazon Comprehend or Amazon Comprehend Medical results and the synthesized entity locations inference to JSON files and saves them in the ML inferences S3 bucket.
11. The Lambda function notifies the calling Step Function of success or failure.
    a. If the text extraction succeeds, this step is complete.
    b. If the text extraction fails, the Step Functions state machine publishes the event to an Amazon SQS dead-letter queue, configured with a default retention period of four days.
12. The solution publishes the success or failure event to the custom event bus, which invokes the workflow orchestrator Lambda function to create the next workflow event.

Redaction workflow

The redaction workflow irreversibly redacts text contained in processed documents (shown as black boxes in the UI).

The redaction workflow is unique from the other workflows in the following ways:

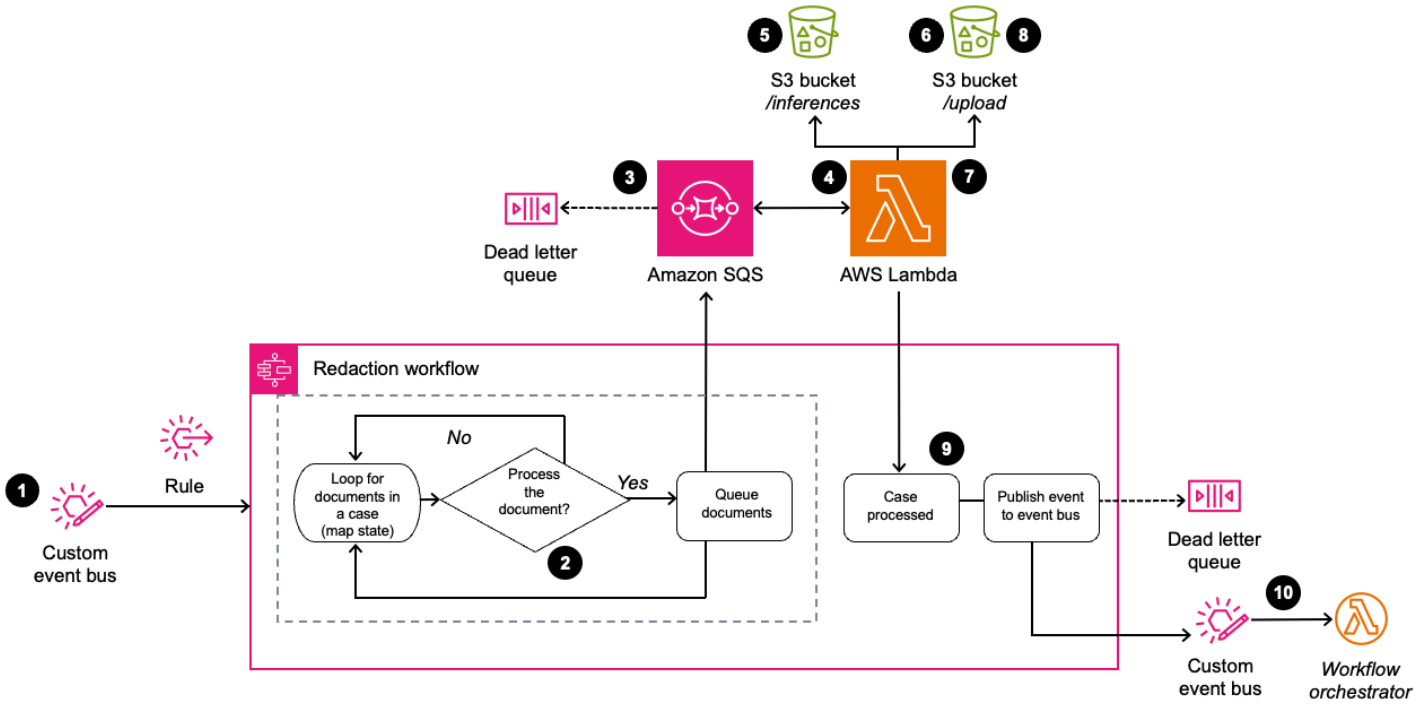
- Includes two separate Lambda functions with shared backing code:
  - One is invoked by the Step Functions workflow as part of a sequence of workflows defined in the workflow configuration.
  - One is manually invoked on processed documents in a case through a REST API.

  These Lambda functions are implemented with the Java runtime.
- Uploads a redacted document to Amazon S3 rather than storing an inference.

- Doesn't interact with the case management store in DynamoDB.

> **Note**
>
> Although this workflow has an option to start from the UI application, to redact specific content (where it is limited to specific entity or phrase in a single request), you can invoke it as a standalone API invocation, with no human interaction or UI. Standalone API invocation supports both phrase redaction and redacting entities from multiple entity types and from entity detection inferences, such as PII and PHI in a single API invocation.

UI workflow



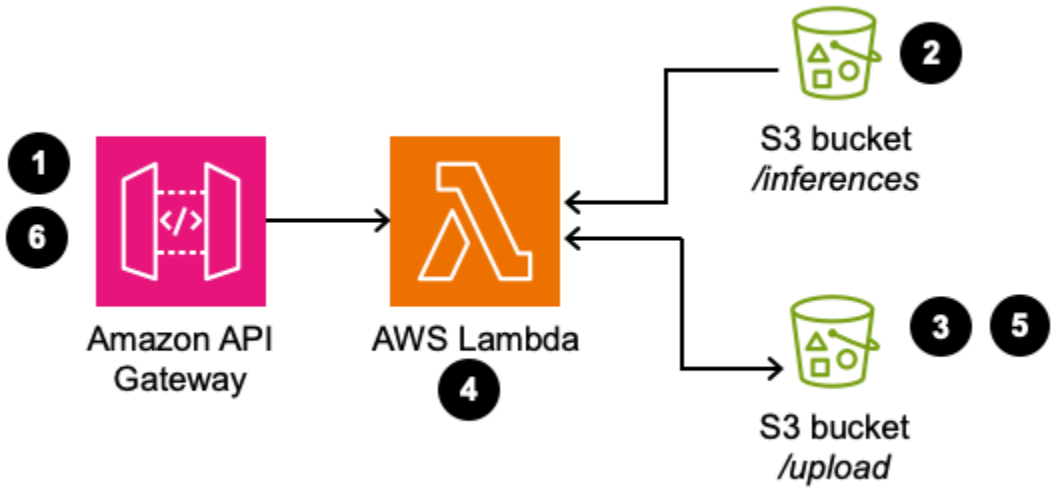
UI redaction workflow

The process flow for the redaction workflow within the UI is as follows:

1. An EventBridge custom event bus invokes a Step Functions state machine.
2. Based on the content of the event, the state machine determines whether the workflow should process each document.

3. An Amazon SQS queue pushes a message with eligible documents and metadata information (for example, the document location in Amazon S3 or the AWS API to use for analysis).
4. A Lambda function consumes the messages from the Amazon SQS queue.
5. The Lambda function retrieves all entity detection inferences available for the given document from the ML inferences S3 bucket.
6. The Lambda function retrieves the original document from the Documents S3 bucket.
7. The Lambda function irreversibly redacts all entities contained in the retrieved inferences files from the document (shown as black boxes over the text).
8. The Lambda function uploads both the redacted document and original document to the Documents S3 bucket. The redacted document includes -redacted appended to the filename.
9. The Lambda function notifies the calling Step Functions of success or failure.
   a. If the text extraction succeeds, this step is complete.
   b. If the text extraction fails, the Step Functions state machine publishes the event to an Amazon SQS dead-letter queue, configured with a default retention period of four days.
10. The solution publishes the success or failure event to the custom event bus.

API workflow



API redaction workflow

The API-based redaction workflow is more powerful. You can use the provided API to:

- Redact specified entities, on specified pages, from the available entity detection inferences.
- Redact specific phrases which are not part of any entity detection inference on specified pages.

> **Important**
>
> You must run the entity detection workflow before running the API redaction workflow.

The process flow for the API redaction workflow is as follows:

1. An API Gateway request invokes a Lambda function.
2. If the API Gateway request specifies entities to redact, the Lambda function retrieves the specified entity detection location inferences for the document from the ML inferences S3 bucket. If the API Gateway request specifies phrases to redact, the Lambda function retrieves the text extraction inference for the document from the ML inferences S3 bucket.
3. The Lambda function retrieves the original document from the Documents S3 bucket.
4. The Lambda function irreversibly redacts the entities contained in the API Gateway request from the document (shown as black boxes over the text).
5. The Lambda function uploads the both the redacted document and original document to the Documents S3 bucket. The redacted document includes `-redacted` appended to the filename.
6. The Lambda function sends an HTTP response to API Gateway based on the outcome.

For details about the expected API request body format, see API reference.

> **Note**
>
> The solution can only store one redacted version of a document at a time. Future runs of the redaction workflow will overwrite previously-redacted documents. If you want to retain previous redactions, enable versioning in the S3 bucket.

How the solution works

You can customize the way this solution processes documents. The solution orchestrates ML inferences, using AWS AI services and their pre-trained models, to extract content and automate

processing. The solution's XML-based configuration provides customization for the orchestration workflow for respective use cases.

Workflows

As shown in the architecture diagram, the solution deploys three workflows: text extraction, entity detection, and redaction. The workflow orchestrator Lambda function orchestrates the order and method of processing uploaded documents using any of these workflows.

Workflow configurations

The solution stores its application workflow configurations in the `workflow-config` DynamoDB table. The solution uses these workflow configurations to:

- Set the number and details of documents required to be uploaded to start processing
- Set the workflows that each document needs to be processed with

The solution creates these DynamoDB table records when the application is deployed. To create the tables, the solution uses configuration JSON files available in the `workflow-config` directory at the root of the application source code.

To add a new configuration, you can clone an existing record in the `workflow-config` DynamoDB table. You can add these records after deployment by signing in to the DynamoDB Console. The format of this configuration file is described below.

You can select which workflow configuration to use by setting the **WorkflowConfigName** parameter during deployment. The workflow orchestrator Lambda function uses this parameter input as the key to retrieve the desired configuration form the `workflow-config` DynamoDB table. This parameter has a default value of `default`.

The following JSON object shows a sample workflow configuration file. During deployment, the solution serializes configuration files such as these into DynamoDB record data and added to the `workflow-config` table. The key of the table corresponds to the **Name** parameter of the configuration file.

```
{
 "Name": "textractToEntity",
 "WorkflowSequence": [
   "textract",
   "entity-standard"
  ],
```

```
 "MinRequiredDocuments": [
    {
     "DocumentType": "generic",
     "FileTypes": [
       ".pdf",
       ".png",
       ".jpeg",
       ".jpg"
      ],
     "RunAmazon TextractAnalyzeAction": false,
     "MaxSize": "5",
     "WorkflowsToProcess": [
       "textract"
      ]
    },
    {
     "DocumentType": "receipt",
     "FileTypes": [
       ".pdf",
       ".png",
       ".jpeg",
       ".jpg"
      ],
     "RunAmazon TextractAnalyzeAction": true,
     "AnalyzeDocFeatureType": ["TABLES", "FORMS", "SIGNATURES"],
     "MaxSize": "5",
     "WorkflowsToProcess": [
       "entity-standard"
      ]
    }
  ]
}
```

The following table describes the details of the configuration.

Parameter	Type	Description	Supported values
Name	String	Name of the workflow configuration. This corresponds to the **WorkflowConfigName**	Any

Parameter	Type	Description	Supported values
		CloudFormation parameter required during deployment.	
WorkflowSequence	Array<String>	The sequence of the document processing workflows to run on an uploaded document, in the order described. **Note** The solution follows the order of items in this list.	• `textract` • `entity-standard` • `entity-pii` • `entity-medical` • `redaction`
MinRequiredDocuments	Array<Map>	This list map object describes the types of documents along with their specs required to execute this workflow. The number of items in this list indicate the number of documents required. The details of the map are described in the following section.	See the following table

The **MinRequiredDocuments** parameter in the configuration file is a list of the required documents to create a workflow. Each item in this list corresponds to the configuration of a single document.

Workflow processing starts only once all of the required types of documents are uploaded to a case.

Parameter	Type	Description	Supported values
DocumentType	String	The user-ascertained type of the uploaded document. Based on the document type, the solution runs the correspon ding Amazon Textract analyze action. There are three textract actions to analyze a document: • `AnalyzeID` for identity documents • `AnalyzeEx pense` for expense-related documents, such as receipts • `AnalyzeDo cument` for generic documents from which key-value pairs are extracted	This solution supports the following types of documents • `driving-l icense` • `passport` • `receipt` • `invoice` • `vaccination- card` • `paystub` • `loan-info rmation` • `health-in surance-card` • `generic` (any other type not listed above)
FileTypes	Array<string>	File type of the uploaded document	• `jpeg` • `jpg` • `png` • `tiff` • `pdf`

Parameter	Type	Description	Supported values
MaxSize	Integer	Maximum file size in megabytes (MB) of a single uploaded document. **Note** The solution has a maximum page limit of 15 pages. This limit is set to ensure that synchrono us Amazon Textract operation s can run reliably. Based on your use case, you may choose to customize the limit. Doing so may impact the system's reliability to handle larger file sizes or additional pages.	Up to 5 MBM

Parameter	Type	Description	Supported values
WorkflowsToProcess	Array<string>	This list is a subset of the **WorkflowSequence** parameter described in the previous table. It indicates the type of processing to run on a specific type of document. You can use this parameter to have fine-grained control of the orchestration process.	• `textract` • `entity-standard` • `entity-pii` • `entity-medical` • `redaction`
RunTextractAnalyzeAction	*(Optional)* Boolean	If you set this parameter to `true`, it Amazon Textract `AnalyzeDocument` and `DetectDocumentText` runs for the document.	• `true` • `false`

Parameter	Type	Description	Supported values
AnalyzeDocFeatureType	*(Optional)* Array<string>	The type of features to detect when Amazon Textract `AnalyzeDocument` is called. If you set **RunTextractAnalyzeAction** to true but this value is missing, then the solution uses a default value of `["TABLES", "FORMS", "SIGNATURES"]` . For more information, see AnalyzeDocument in the *Amazon Textract Developer Guide*.	List containing any of: • `TABLES` • `FORMS` • `SIGNATURES`

Plan your deployment

This section describes the cost, security, quotas, and other considerations prior to deploying the solution.

Supported AWS Regions

This solution uses AWS AI services such as Amazon Comprehend, Amazon Textract, and Amazon Kendra and Amazon OpenSearch Serverless, which are not currently available in all AWS Regions. You must launch this solution in an AWS Region where these services are available. For the most current availability of AWS services by Region, see the AWS Regional Services List.

Enhanced Document Understanding on AWS is supported in the following AWS Regions:

Region name	
US East (Ohio)	Asia Pacific (Sydney)
US East (N. Virginia)	Asia Pacific (Singapore)
US West (Oregon)	Europe (Ireland)

When you deploy the solution without Amazon Kendra, and Amazon OpenSearch Serverless,Enhanced Document Understanding on AWS is supported in the following AWS Regions:

Region name	
US East (Ohio)	Asia Pacific (Sydney)
US East (N. Virginia)	Canada (Central)
US West (Oregon)	Europe (Frankfurt)
Asia Pacific (Mumbai)	Europe (Ireland)
Asia Pacific (Seoul)	Europe (London)
Asia Pacific (Singapore)	

Cost

You are responsible for the cost of the AWS services used while running this solution. As of this revision, the cost for running this solution with the default settings in the US East (N. Virginia) Region is approximately **$1,847.28 USD per month**. These costs are for the resources shown in the Sample cost table.

See the pricing webpage for each AWS service used in this solution.

We recommend creating a budget through AWS Cost Explorer to help manage costs. Prices are subject to change. For full details, see the pricing webpage for each AWS service used in this solution.

Sample cost tables

The following tables provides a sample cost breakdown for deploying this solution with the default parameters in the US East (N. Virginia) Region for one month, split by fixed and variable costs.

Fixed costs

AWS service	Dimensions	Cost [USD]
Amazon API Gateway	1,000,000 REST API calls per month, including uploads and downloads, and caching disabled	$3.50
Amazon CloudFront	1,000,000 requests – for 20 GB of data transfer out to the internet and 20GB of data transfer out to the origin	$2.10
Amazon CloudWatch	24 metrics using 5 GB of data ingested for logs and 1 dashboard	$9.72
Amazon Cognito	10,000 active users per month with the advanced security feature and with	$514.25

AWS service	Dimensions	Cost [USD]
	10% users signing in through SAML 2.0 or OpenID Connect (OIDC) federation	
Amazon DynamoDB	2 DynamoDB tables with point-in-time recovery (PITR) enabled: `workflow-config` DynamoDB table with 20 configurations 250 MB `CaseManager` table with 500,000 cases, and 50 reads and 50 writes per second	$0.00 $180.74
Amazon EventBridge	1,000,000 custom events with 1 KB payload	$1.00

AWS service	Dimensions	Cost [USD]
AWS Lambda	1,000,000 requests to Lambda	$1.77
	850,000 requests with 128 MB Lambda memory and 512 MB ephemeral storage to 8 Lambda functions with 1 second average duration	$0.31
	100,000 requests with 192 MB Lambda memory and 512 MB ephemeral storage to 1 Lambda function with 1 second average duration	$4.17
	50,000 requests with 1,024 MB Lambda memory and 512 MB ephemeral storage to 1 Lambda function with 5 second duration	
Amazon S3	~1 TB storage for 200,000 500 KB documents, JSON inferences, and 50,000 redaction-processed documents in Standard S3	$26.32
	1,000,000 (at 500,000 GET + 500,000 POST) requests	
	100 GB data returned by Amazon S3	
Amazon SQS	1,000,000 standard queue requests	$0.00

AWS service	Dimensions	Cost [USD]
AWS Step Functions	20 state transitions for 500,000 workflow requests (10,000,000 state transitions per month)	$250.00
AWS Systems Manager Parameter Store	500,000 parameter store API interactions with 8 standard parameters (standard throughput API interactions enabled)	$25.00
AWS WAF	1,000,000 for 1 web access control list and (ACL) 7 defined rules	$13.00
AWS VPC	2 Nat Gatways and 2 Privateli nks	$72.00
Total Fixed Costs		**$1,103.88**

Variable costs

This solution provides the flexibility to select only the AI-powered service combinations that fit your use case. Consider the following use cases.

Use case 1 - Solution deployed with the intent to analyze expenses with Amazon Textract and Amazon Comprehend that detects named and PII entities.

AWS service	Dimensions	Cost [USD]
Amazon Comprehend	6,000 documents with 5 pages per document = 30,000 pages with an average of 1,700 characters per page; 100 characters per unit = x units for synchronous	$10.20 $10.20

AWS service	Dimensions	Cost [USD]
	Amazon Comprehend NERe Amazon Comprehend PII detection	
Amazon Textract	30,000 pages of `DetectText` API 10,000 pages of `AnalyzeText` API for documents containing tables and forms 10,000 pages of `AnalyzeExpense` API	$45.00 $650.00 $100.00
Total combination cost		**$815.40**
+ Fixed costs ($1,031.88)		**$1,847.28**

Use case 2 - Solution deployed with the intent to detect and redact PHI entities from Amazon Textract and Amazon Comprehend Medical.

AWS service	Dimensions	Cost [USD]
Amazon Comprehend Medical	6,000 documents with 5 pages per document = 30,000 pages with an average of 1,700 characters per page; 100 characters per unit = x units for synchronous Amazon Comprehend Medical PHI detection	$714.00
Amazon Textract	30,000 pages of `DetectText` API	$45.00

AWS service	Dimensions	Cost [USD]
Total combination cost		**$759.00**
+ Fixed costs ($1,031.88)		**$1,790.88**

Use case 3 - Solution deployed for identifying and matching patients for clinical trials based on medical criteria (entities) in clinical notes and research forms using Amazon Kendra Enterprise Edition, Amazon OpenSearch Serverless, Amazon Textract, and Amazon Comprehend Medical.

AWS service	Dimensions	Cost [USD]
Amazon Kendra	8,000 queries a day, up to 100,000 documents with Amazon Kendra Enterpris e Edition and up to 50 data sources (Enterprise edition default storage and query capacities)	$1,008.00
Amazon OpenSearch Serverless	Hot ephemeral storage for 120 GiB of index data, per OCU and supports up to 10 TiB of hot data per index in a time series collection.	$691.20
Amazon Comprehend Medical	2,000 documents with 5 pages per document = 10,000 pages with an average of 1,700 characters per page; 100 characters per unit = 85,000 units Amazon Comprehend Medical NERe API	$1,700.00

AWS service	Dimensions	Cost [USD]
Amazon Textract	30,000 pages of `DetectText` API 10,000 pages of `AnalyzeText` API for documents containing tables and forms 10,000 pages of `AnalyzeID` API 10,000 pages of `AnalyzeExpense` API	$45.00 $650.00 $100.00 $100.00
Total combination cost		**$4,294.20**
+ Fixed costs ($1,031.88)		**$5,326.08**

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared responsibility model reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information about AWS security, visit AWS Cloud Security.

IAM roles

IAM roles allow customers to assign granular access policies and permissions to services and users on the AWS Cloud. This solution creates IAM roles that grant the solution's Lambda functions access to create AWS Regional resources.

Amazon CloudFront

This solution deploys a web frontend hosted in an Amazon S3 bucket. To help reduce latency and improve security, this solution includes a CloudFront distribution with an origin access identity, which is a CloudFront user that provides public access to the solution's website bucket contents.

For more information, see Restricting Access to Amazon S3 Content by Using an Origin Access Identity in the *Amazon CloudFront Developer Guide*.

We recommend deploying AWS WAF in front of the CloudFront distribution for enhanced security. This is not enabled by default because AWS WAF for CloudFront can't be deployed in all Regions supported by this solution. See Using AWS WAF to control access to your content in the *Amazon CloudFront Developer Guide* for details.

Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the services implemented in this solution. For more information, see AWS service quotas.

To view the service quotas for all AWS services in the documentation without switching pages, view the information in the Service endpoints and quotas page in the PDF instead.

AWS CloudFormation quotas

Your AWS account has CloudFormation quotas that you should be aware of when launching the stack in this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, see AWS CloudFormation quotas in the in the *AWS CloudFormation User's Guide*.

Regulatory requirements

If you need the solution to follow specific regulatory compliance requirements, such as General Data Protection Regulation (GDPR) and HIPAA, review them to ensure that the solution meets those requirements before deploying it.

Deploy the solution

This solution uses CloudFormation templates and stacks to automate its deployment. The CloudFormation template specifies the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources that are described in the template.

Deployment process overview

Before you launch the solution, review the cost, architecture, security, and other considerations discussed in this guide.

Time to deploy: Approximately:

- 30-45 minutes with the Amazon Kendra Index
- 15 minutes without the Amazon Kendra Index

> **Important**
>
> This solution includes an option to send anonymous operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered though this survey. Data collection is subject to the AWS Privacy Notice.
>
> To opt out of this feature, download the template, modify the AWS CloudFormation mapping section, and then use the AWS CloudFormation console to upload your updated template and deploy the solution. For more information, see the Anonymized data collection section of this guide.

AWS CloudFormation template

You can download the CloudFormation template for this solution before deploying it.

View template

DocUnderstanding.template – Use this template to launch the solution and all associated

components. The default configuration deploys the core and supporting services found in the AWS services in this solution section, but you can customize the template to meet your specific needs.

> **Note**
>
> CloudFormation resources are created from AWS CDK constructs.

This CloudFormation template deploys Enhanced Document Understanding on AWS in the AWS Cloud.

Launch the stack

Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

1. Sign in to the AWS Management Console and select the button to launch the `DocUnderstanding.template` CloudFormation template.

   Launch solution

2. The template launches in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.

   > **Note**
   >
   > This solution optionally uses the Amazon Kendra and OpenSearch Serverless, which are not currently available in all AWS Regions. If you want to deploy the solution with natural language or keyword search, you must launch this solution in an AWS Region where Amazon Kendra and Aamzon OpenSearch Serverless are available. For the most current availability by Region, see the AWS Regional Services List.

3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, see IAM and AWS STS Quotas in the *AWS Identity and Access Management User Guide*.

5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
DeployKendraIndex	No	This parameter tells the solution whether to create and deploy an Amazon Kendra index to add uploaded documents. Setting this to `Yes` will also enable the search functionality on the UI through the API. Refer to the API reference section for details on how to invoke.
DeployOpenSearch	No	This parameter tells the solution whether to create and deploy an Amazon OpenSearch Serverless collection to add uploaded documents. Setting this to `Yes` will also enable the search functionality on the UI through the API. Refer to the API reference section for details on how to invoke.
NotificationSubscriptionEmail	*Optional input*	The email address that you want the solution to use for notification emails. The solution creates a Cognito user credential and sends instructions on how to access the application after deployment.

Parameter	Default	Description
WorkflowConfigName	`default`	Name of the workflow configuration for the orchestrator to run for document processing. See How the solution works for more information.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review and create** page, review and confirm the settings. Select the boxes acknowledging that the template:
   - Creates IAM resources
   - Might require the `CAPABILITY_AUTO_EXPAND` capability
9. Choose **Submit** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a CREATE_COMPLETE status in 30-45 minutes.
10. This solution also deploys sample documents that you can use for processing and exploring features of the solution.

> **Note**
>
> In addition to the primary Lambda functions, this solution includes a solution helper Lambda function, which runs only during initial configuration or when resources are updated or deleted. You must not delete the solution helper function, as it is necessary to manage associated resources.

The solution deploys a CloudWatch dashboard that charts custom and AWS-provided metrics to provide insights into the deployed application's usage. You can find the custom dashboard by signing in to the CloudWatch console and selecting **Dashboards** from the navigation menu.

Post-deployment configuration

This section provides recommendations for configuring the solution after deployment.

Amazon S3 bucket versioning, lifecycle policies and cross-Region replication

This solution uses Amazon S3 to store:

- Documents uploaded by users.
- Results from processing documents, including but not limited to inferences and redacted documents.
- Other assets (see AWS services in this solution).

This solution doesn't enforce lifecycle configurations on the buckets it creates. We recommend:

- Setting lifecycle configurations for production deployments. See Setting lifecycle configuration on a bucket for details.
- Enabling versioning and cross-region replication for S3 buckets based on the use case for which the solution is deployed.

Amazon DynamoDB backups

This solution uses DynamoDB for several purposes (see AWS services in this solution). The solution doesn't enable backups for the tables it creates. We recommend enabling point-in-time recovery for DynamoDB, creating a backup of this feature for production deployments. See Backing up a DynamoDB table and Using AWS Backup for DynamoDB for details.

Amazon CloudWatch dashboard and alarms

The solution deploys a custom dashboard in CloudWatch to render charts from custom published metrics and AWS service metrics. We recommend creating CloudWatch alarms and adding notifications based on the use case for which the solution is deployed.

Scaling with Amazon Kendra

This solution provides the ability to use Amazon Kendra to perform NLP-powered intelligent search across the uploaded documents. You can increase the capacity of Amazon Kendra using the following CloudFormation parameters for larger workloads:

Parameter	Default	Description	Capacity options
QueryCapacityUnits	`0`	The amount of extra query capacity for an index and GetQuerySuggestions capacity. An additional capacity unit for an index provides approximately 8,000 queries per day.	The solution allows you to choose between `0` and 1 values for this parameter
StorageCapacityUnits	`0`	The amount of extra storage capacity for an index. A single capacity unit provides 30 GB of storage space or 100,000 documents, whichever is reached first.	The solution allows you to choose between `0` and 5 values for this parameter
Kendra Index Edition	`DEVELOPER_EDITION`	Amazon Kendra provides Developer and Enterprise Editions to create indexes. For more information about the differences between Amazon Kendra Editions,	The solution allows you to choose between `DEVELOPER_EDITION` and `ENTERPRISE_EDITION`

Parameter	Default	Description	Capacity options
		see Amazon Kendra pricing.	

To modify the values of these CloudFormation parameters, select the appropriate values at the time of stack deployment. For more information on query and storage capacity units, see Adjusting capacity.

> **Note**
>
> If the solution is not deployed with Amazon Kendra, the search feature is not available.

Custom web domains with TLS v1.2 or higher certificates

The solution deploys a UI using CloudFront. CloudFront's domain doesn't enforce TLS v1.2 or higher certificates. We recommend creating a custom domain using Amazon Route 53, creating a certificate using AWS Certificate Manager, or using an existing certificate if your organization has one.

Refer to the Amazon Route 53 Developer Guide for registering a new domain name.

Additional security considerations

Based on the use case for which you deploy the solution, review the following security recommendations:

- **Customer managed AWS KMS encryption keys** - The solution uses AWS managed AWS KMS keys by default, as these are available at no additional cost. Review your use case to determine if you need to update the solution to use customer managed AWS KMS keys.
- **API Gateway throttling rules** - The solution deploys with default throttling rules on API Gateway. Based on your use case and expected transaction volumes, we recommend that you configure throttling for the APIs. See Throttle API requests for better throughput in the Amazon API Gateway Developer Guide for details.
- **Enabling AWS CloudTrail** - As a recommended security practice, consider enabling AWS CloudTrail in the AWS account where the solution is deployed to log API calls in the AWS account. See the AWS CloudTrail User Guide for more details.

- **AWS WAF with Amazon CloudFront** - To enhance the security of the CloudFront distribution, we recommend configuring AWS WAF with CloudFront after deploying the solution. AWS WAF for CloudFront is available only in the US East (N. Virginia) Region but can be configured with a solution deployed in other AWS Regions.
- **Malware scanning** - If your use case requires that documents be scanned for malicious contents, we recommend integrating the `RequestProcessorDocumentRepo` bucket with a malware scanning solution into the workflow. See Integrating Amazon S3 Malware Scanning into Your Application Workflow with Cloud Storage Security for more information.
- **Drift detection** - We recommend configuring drift detection on CloudFormation stacks to identify and be notified of unintentional or malicious changes to the deployed solution stack. See Implementing an alarm to automatically detect drift in AWS CloudFormation stacks for details.
- **Sensitive documents** - We recommend configuring continuous monitoring services like Amazon Macie to identify if sensitive documents were unintentionally uploaded to S3 buckets, and Amazon GuardDuty to detect any security threats along with the deployed solution.
- **Cognito JSON Web Tokens (JWTs)** – The solution uses Cognito-issued JWTs to authenticate with the REST API endpoints. We configured the solution with a five-minute expiry for ID tokens and access tokens. When a user logs out, their ability to generate new tokens is revoked (refresh token is revoked). However, until the expiry of the current token, any requests to the API endpoint will be successfully authenticated, since they have a valid token. Review the security considerations for your use case and adjust the token validity period.

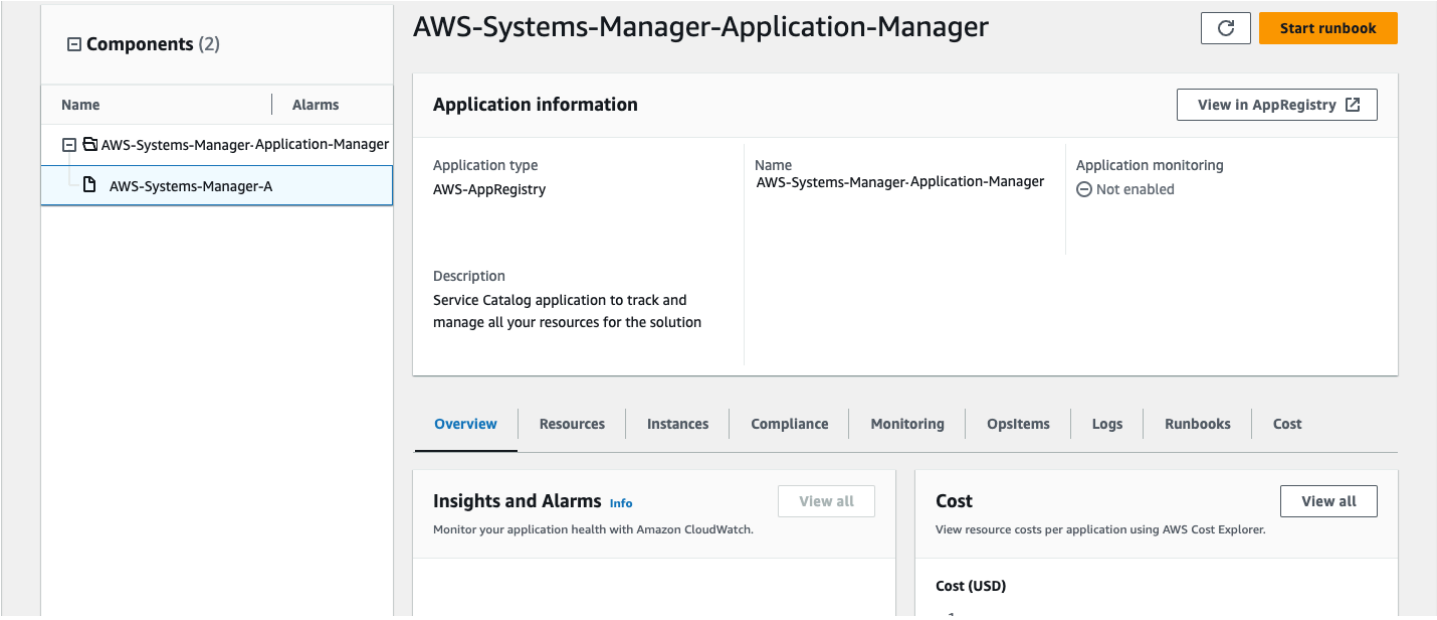
Monitor the solution with Service Catalog AppRegistry

This solution includes a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both Service Catalog AppRegistry and AWS Systems Manager Application Manager.

AWS Systems Manager Application Manager gives you an application-level view into this solution and its resources so that you can:

- Monitor its resources, costs for the deployed resources across stacks and AWS accounts, and logs associated with this solution from a central location.
- View operations data for the resources of this solution (such as deployment status, CloudWatch alarms, resource configurations, and operational issues) in the context of an application.

  The following figure depicts an example of the application view for the solution stack in Application Manager.



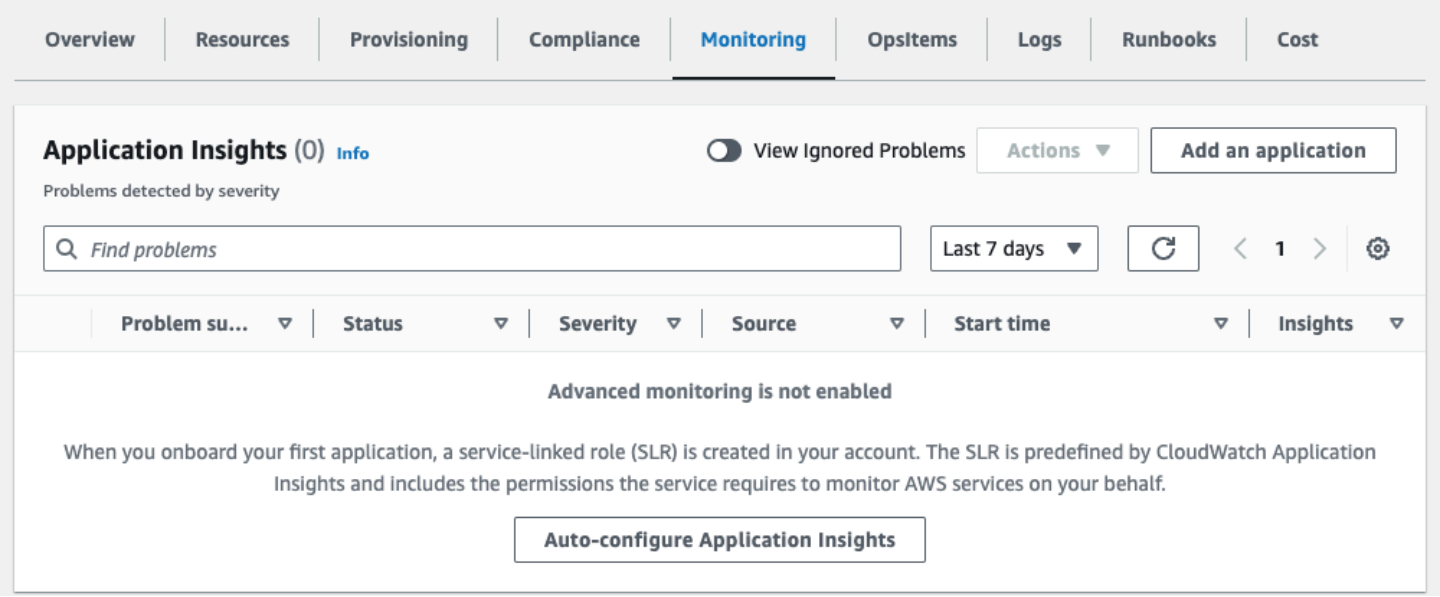
Solution stack in Application Manager

Activate CloudWatch Application Insights

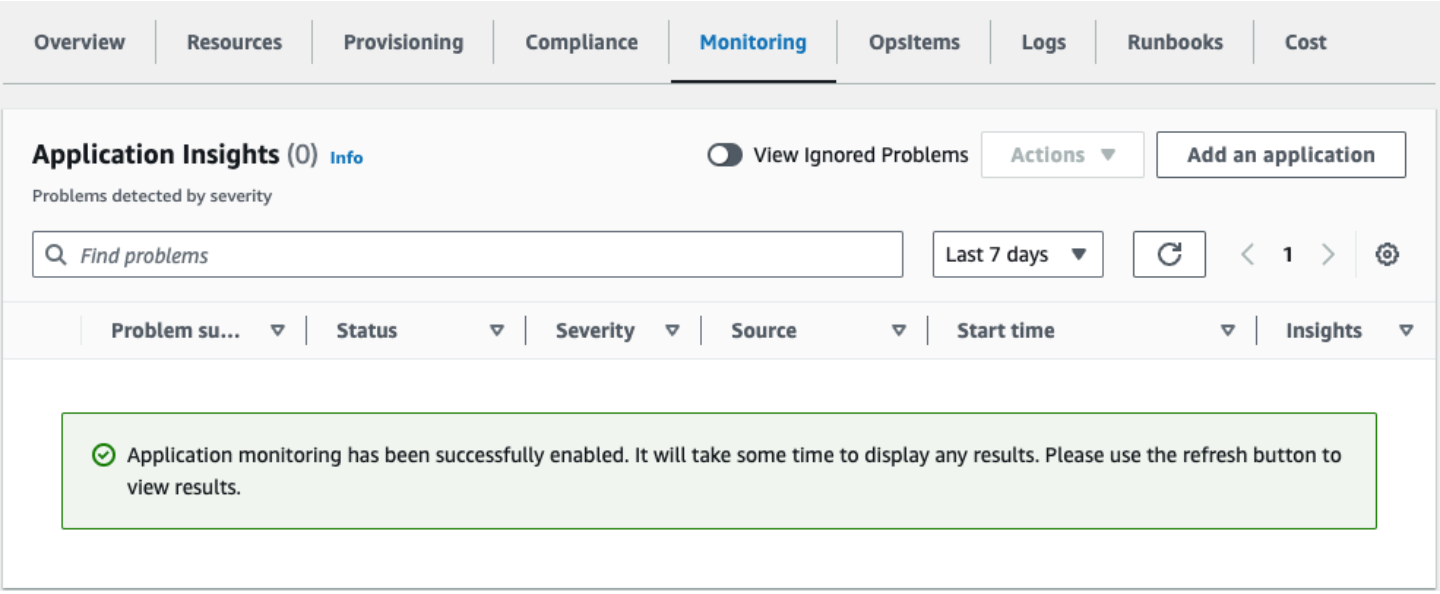
1. Sign in to the Systems Manager console.

2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, search for the application name for this solution and select it.

   The application name will have App Registry in the **Application Source** column, and will have a combination of the solution name, Region, account ID, or stack name.
4. In the **Components** tree, choose the application stack you want to activate.
5. In the **Monitoring** tab, in **Application Insights**, select **Auto-configure Application Insights**.



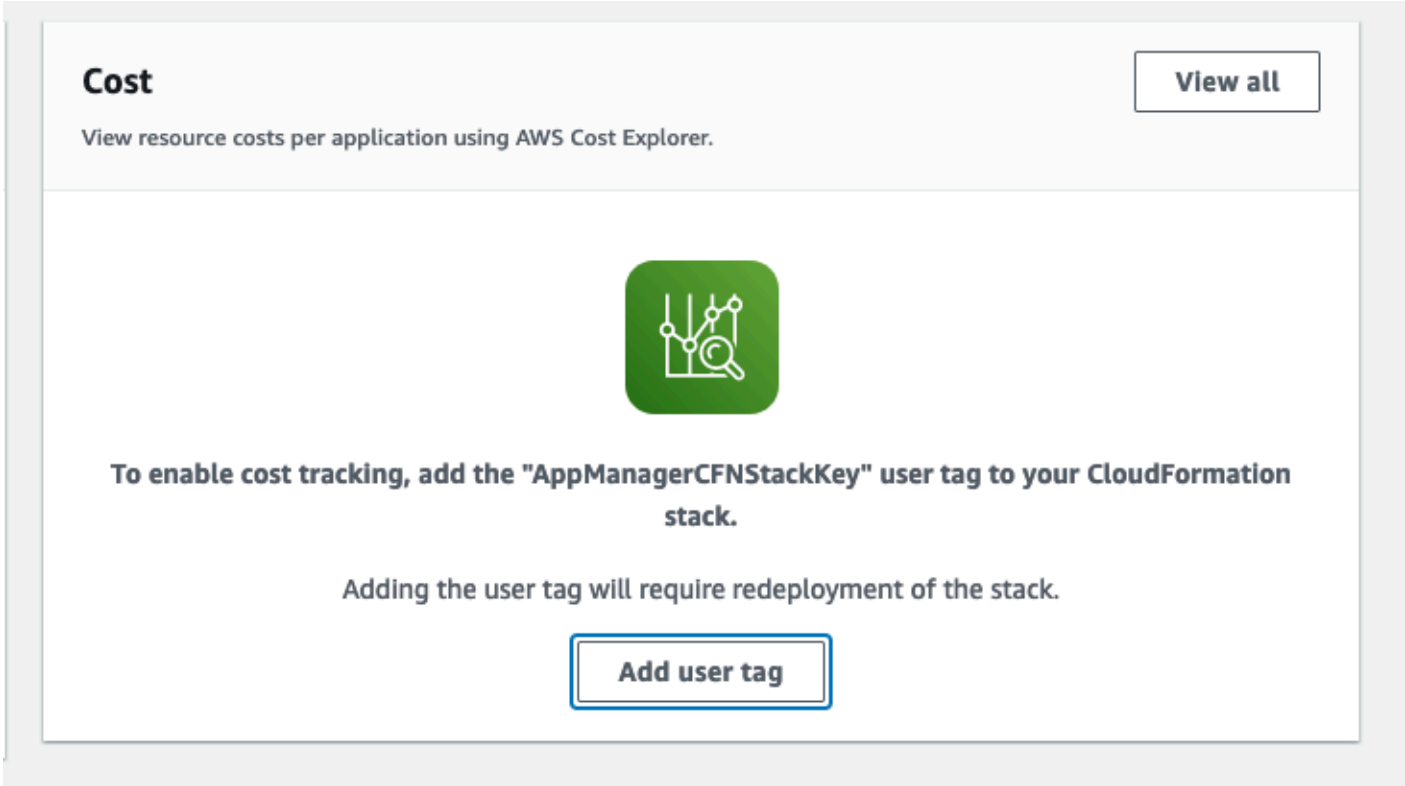
Monitoring for your applications is now activated and the following status box appears:



Confirm cost tags associated with the solution

After you activate cost allocation tags associated with the solution, you must confirm the cost allocation tags to see the costs for this solution. To confirm cost allocation tags:

1. Sign in to the Systems Manager console.
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose the application name for this solution and select it.
4. In the **Overview** tab, in **Cost**, select **Add user tag**.



5. On the **Add user tag** page, enter `confirm`, then select **Add user tag**.

The activation process can take up to 24 hours to complete and the tag data to appear.

Activate cost allocation tags associated with the solution

After you confirm the cost tags associated with this solution, you must activate the cost allocation tags to see the costs for this solution. The cost allocation tags can only be activated from the management account for the organization.

To activate cost allocation tags:

1. Sign in to the AWS Billing and Cost Management and Cost Management console.
2. In the navigation pane, select **Cost Allocation Tags**.
3. On the **Cost allocation tags** page, filter for the `AppManagerCFNStackKey` tag, then select the tag from the results shown.
4. Choose **Activate**.

AWS Cost Explorer

You can see the overview of the costs associated with the application and application components within the Application Manager console through integration with AWS Cost Explorer. Cost Explorer helps you manage costs by providing a view of your AWS resource costs and usage over time.

1. Sign in to the AWS Cost Management console.
2. In the navigation menu, select **Cost Explorer** to view the solution's costs and usage over time.

Troubleshooting

This section provides troubleshooting instructions for deploying and using the solution.

If these instructions don't address your issue, see the Contact AWS Support section for instructions on opening an AWS Support case for this solution.

Problem: Document is not processed

After you upload a document, the solution doesn't process it.

Resolution

Check the workflow configuration that is currently deployed in the CloudFormation template or in the DynamoDB table containing the workflow configuration. If the configuration specifies that a case requires more than one document, the processing of all documents begins after you upload the minimum set of documents.

Problem: Document processing fails

After you upload a document, the processing fails and you receive a 'Failed' notification email.

Resolution

To check the root cause of failure:

1. Sign in to the AWS Management Console.
2. In the Region selector, select the Region where the solution is deployed.
3. Sign in to the AWS Step Functions console. Based on your deployment, there could be up to three state machine definitions (for Text Extraction, Entity Detection, and Redaction).
4. Select the **Failed** column for the corresponding state machine definition to see additional details.
5. Select the stage where it failed and check **Output** of that stage to see the error.

Problem: Download Redacted Document fails

When you choose **Download Redacted Document** while viewing the document analysis, no document is downloaded or the browser freezes.

Resolution

1. Confirm whether the browser is prompting to **Allow pop-ups**.
2. If yes, as a one-time choice, select **Always allow** for this website.

> **Note**
>
> The exact text of the option varies by browser.

3. Attempt to choose **Download Redacted Document** again.

Contact Support

If you have AWS Developer Support, AWS Business Support, or AWS Enterprise Support, you can use the Support Center to get expert assistance with this solution. The following sections provide instructions.

Create case

1. Sign in to Support Center.
2. Choose **Create case**.

How can we help?

1. Choose **Technical**.
2. For **Service**, select **Solutions**.
3. For **Category**, select **Other Solutions**.
4. For **Severity**, select the option that best matches your use case.
5. When you enter the **Service**, **Category**, and **Severity**, the interface populates links to common troubleshooting questions. If you can't resolve your question with these links, choose **Next step: Additional information**.

Additional information

1. For **Subject**, enter text summarizing your question or issue.
2. For **Description**, describe the issue in detail.
3. Choose **Attach files**.
4. Attach the information Support needs to process the request.

Help us resolve your case faster

1. Enter the requested information.
2. Choose **Next step: Solve now or contact us**.

Solve now or contact us

1. Review the **Solve now** solutions.
2. If you can't resolve your issue with these solutions, choose **Contact us**, enter the requested information, and choose **Submit**.

Uninstall the solution

You can uninstall the Enhanced Document Understanding on AWS solution from the AWS Management Console or by using the AWS Command Line Interface (AWS CLI). You must manually delete the S3 bucket, DynamoDB tables, and CloudWatch logs created by this solution. AWS Solutions don't automatically delete S3 buckets, DynamoDB tables, and CloudWatch logs in case you stored data that you want to retain.

Using the AWS Management Console

1. Sign in to the AWS CloudFormation console.
2. On the **Stacks** page, select this solution's installation stack.
3. Choose **Delete**.

Using AWS Command Line Interface

Determine whether the AWS CLI is available in your environment. For installation instructions, see What Is the AWS Command Line Interface in the *AWS CLI User Guide*. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

Deleting the Amazon S3 buckets

To prevent accidental data loss, we configured this solution to retain the solution-created S3 bucket if you decide to delete the CloudFormation stack. After uninstalling the solution, you can manually delete this S3 bucket if you don't need to retain the data. Follow these steps to delete the Amazon S3 bucket.

1. Sign in to the Amazon S3 console.
2. Choose **Buckets** from the navigation pane.
3. Locate the *<stack-name>* S3 buckets.
4. Select each S3 bucket and choose **Delete**.

To delete the S3 bucket using AWS CLI, run the following command for each S3 bucket:

```
$ aws s3 rb s3://<bucket-name> --force
```

Deleting the Amazon DynamoDB tables

To prevent accidental data loss, we configured this solution to retain the DynamoDB tables if you decide to delete the CloudFormation stack. After uninstalling the solution, you can manually delete the DynamoDB tables if you don't need to retain the data. Follow these steps to delete the DynamoDB tables.

1. Sign in to the Amazon DynamoDB console.
2. Choose **Tables** from the navigation pane.
3. Select the *<stack-name>* tables and choose **Delete**.

To delete the DynamoDB tables using AWS CLI, run the following command:

```
$ aws dynamodb delete-table <table-name>
```

Deleting the CloudWatch Logs

To prevent accidental data loss, we configured this solution to retain the CloudWatch logs if you decide to delete the CloudFormation stack. After uninstalling the solution, you can manually delete the logs if you don't need to retain the data. Follow these steps to delete the CloudWatch logs.

1. Sign in to the Amazon CloudWatch console.
2. Choose **Log Groups** from the navigation pane.
3. Locate the log groups created by the solution.
4. Select one of the log groups.
5. Choose **Actions** and then choose **Delete**.

Repeat the steps until you have deleted all the solution log groups.

Deleting the CloudWatch Logs

Unfortunately, OpenSearch Serverless does not currently support taking snapshots. Deleting this solution will delete the provisioned OpenSearch serverless collection as well, please make sure all data is exported before the deletion.

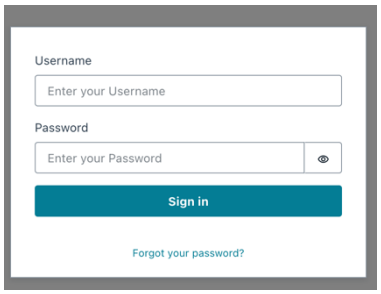
Use the solution

This section provides a user guide for using this AWS Solution.

Sign in to the UI

This section provides step-by-step instructions for signing into the solution UI.

1. Launch the UI in your web browser by following these steps:
   - Sign in to the AWS CloudFormation console.
   - On the **Stacks** page, select this solution's UI stack (for example, *<stack-name>*-`WebAppS3UINestedStackS3UINestedStackResourceXXXX`).
   - Select the **Outputs** tab, and choose the hyperlink value for the `WebUrl` key.



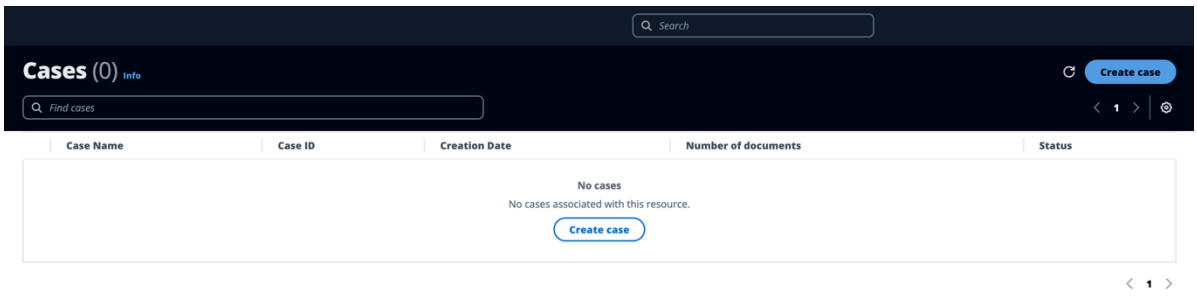
2. Enter the credentials that you received in the email from `<no-reply@verificationemail.com>`.
3. At the first sign in, you are prompted to change your password. Enter a new password and sign in with the new credentials.

Upload a document

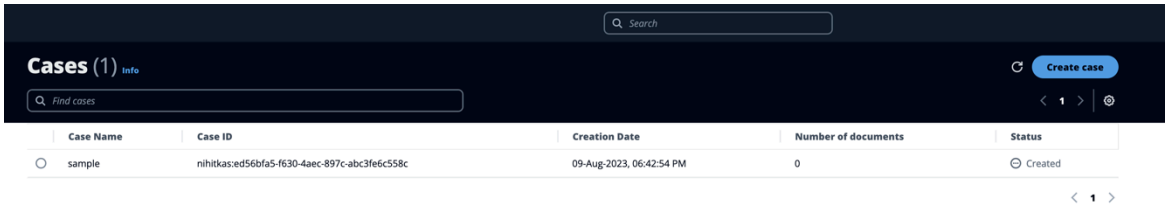
This section provides step-by-step instructions for uploading a document to the solution UI.

1. Sign in to the UI.

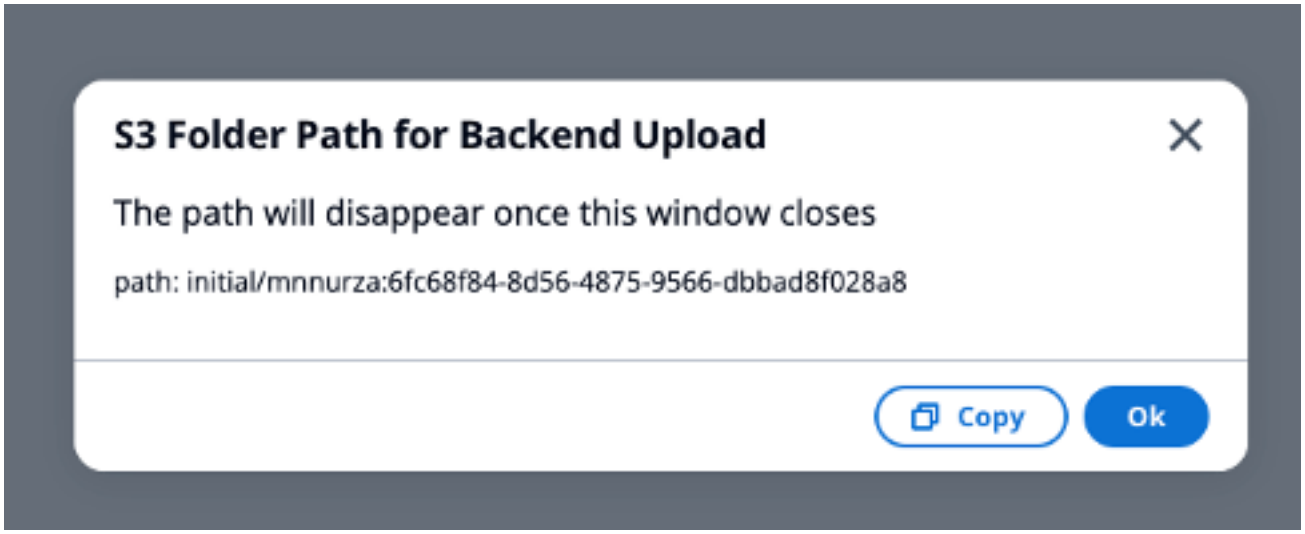
   The UI provides a list of cases (if there are existing cases).



2. Choose **Create case**.
3. Enter a case name and choose **Create case**.

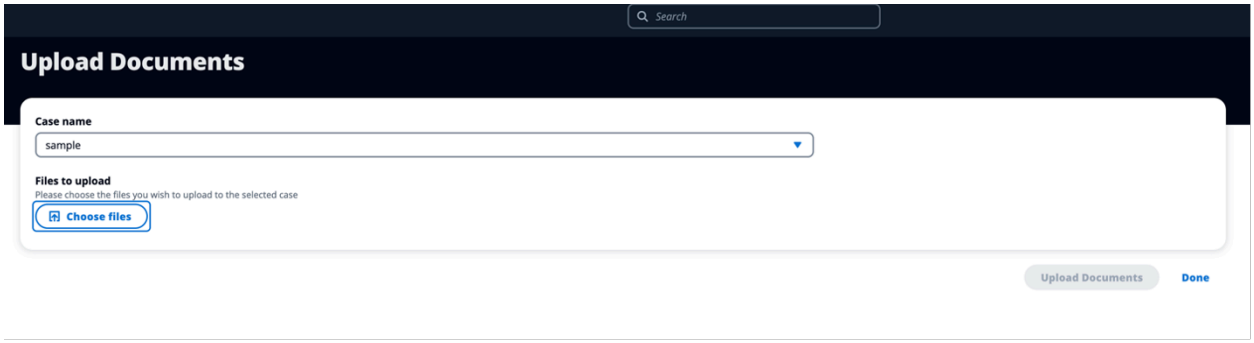


4. You have the choice to enable the document upload through the UI or through the backend upload:
   - If you choose to enable the Backend upload, proceed to Step 5.
   - Otherwise, skip to Step 6.
5. A window appears at the bottom of the page to allow document uploads.
   - Copy the path using the copy button, (Note: If you close out of this window it will disappear from the UI and cannot be recovered from the UI)



6. Using a program, script or AWS CLI upload to AWS s3 using that path and these required tags:
   a. `userId`
   b. `fileNameBase64EncodedTag`
      - The file name will need to be encoded to base64 for this to work.
      - For more information see tag restrictions.
   c. `documentType`
7. The created case appears in the list. The **Number of documents** column shows `0`documents.

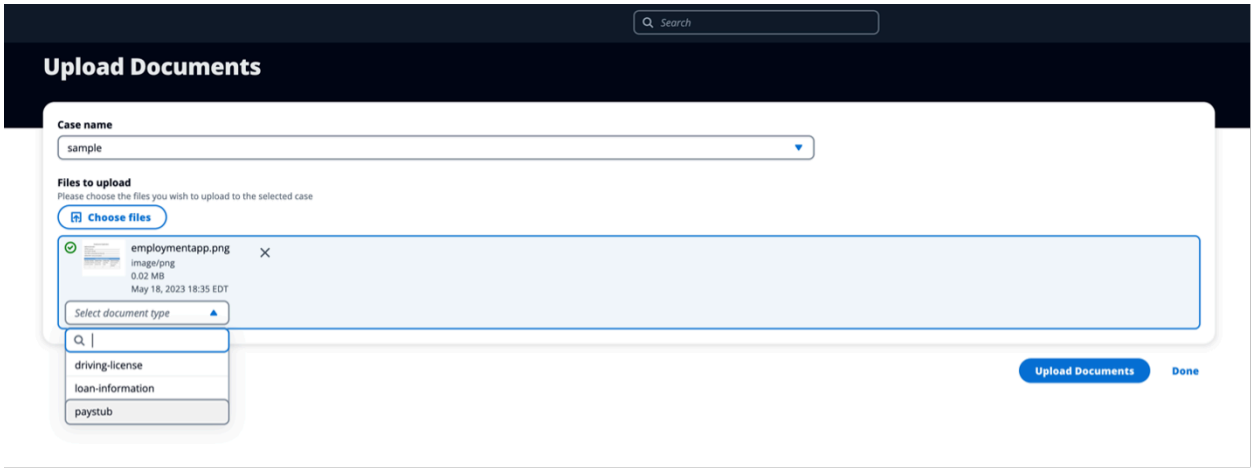
8. Select the case.
   - A window appears at the bottom of the page to allow document uploads.
9. Choose **Upload Documents**, then choose **Done**.



10Select the files that you want to upload.

11. Choose **Select the document type** and select the type that best applies to the file.

> **Note**
>
> The list of document types to select is based on the configuration selected when deploying the solution as a CloudFormation stack. . See How the solution works for details.



12. Choose **Upload Documents**, then choose **Done**.

After you upload the required number of documents, the processing workflows start.

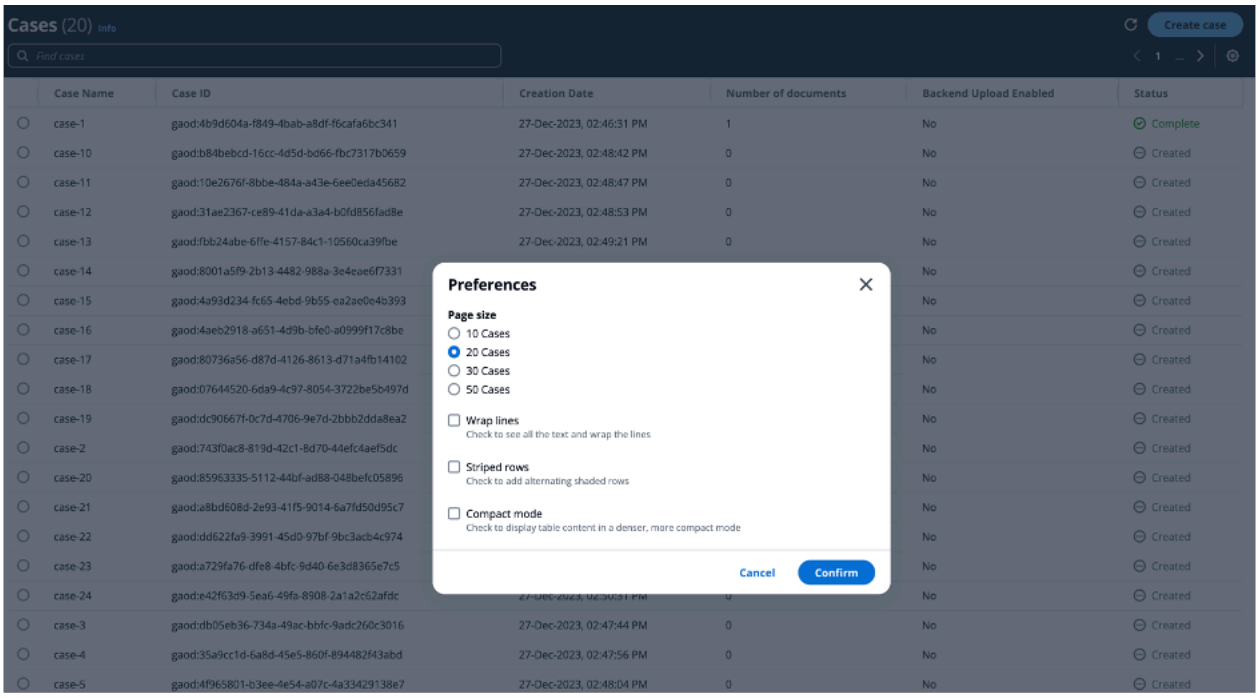
> **Note**
>
> The list of document types to select is based on the configuration selected when deploying the solution as a CloudFormation stack. . See How the solution works for details.

View Paginated Cases

This section provides step-by-step instructions for using the components in the solution UI to see paginated cases.

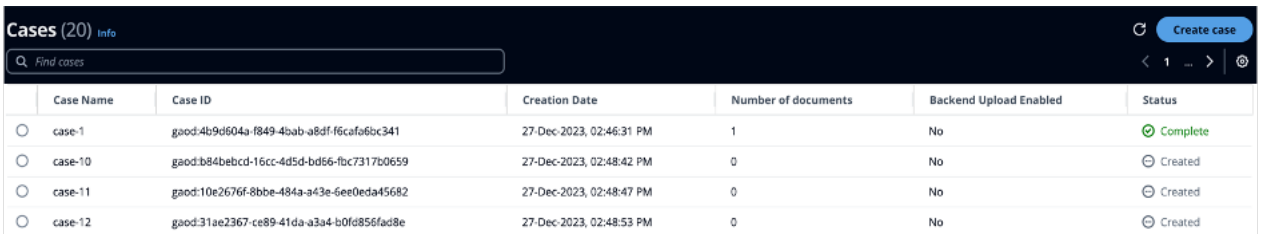
1. Sign in to the UI.

2. Click gear icon, and configure preferences on how many cases each page shows up. This step is not required as UI uses 20 cases per page as a default setting.



3. The UI retrieves one page at a time, and it retrieves next 20 cases (if page size preference is overridden) when the next page arrow is clicked.

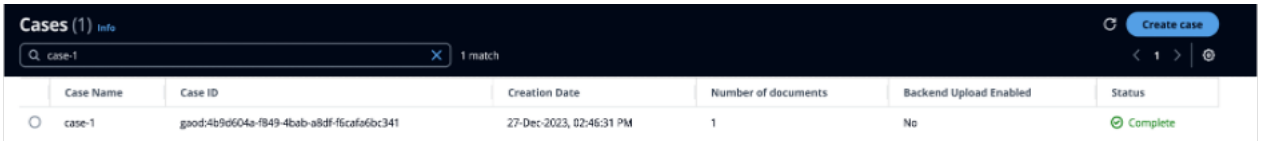
> **Note**
>
> Due to Cloudscape limitation, the page index always shows as one.



4. Use next page arrow to flip pages until the wanted case shows up. Returned cases are sorted by case name.

> **Note**
>
> Current UI does not allow moving back to previous pages, instead it only allows moving to next page.

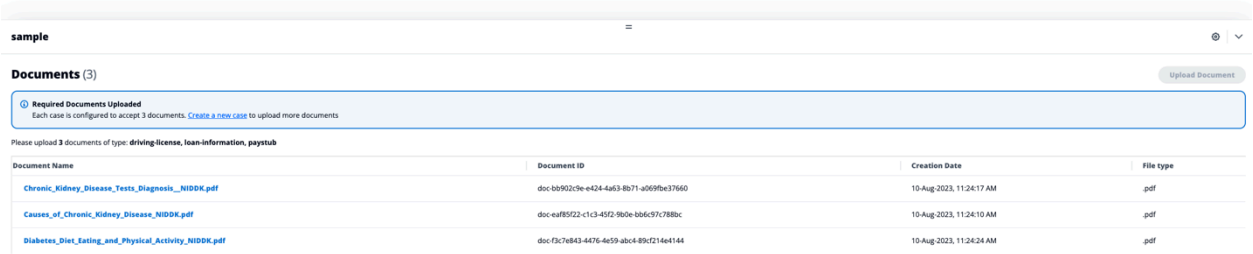
5. Search case by typing case name in input box, UI returns the case which exactly matches the typped case name.

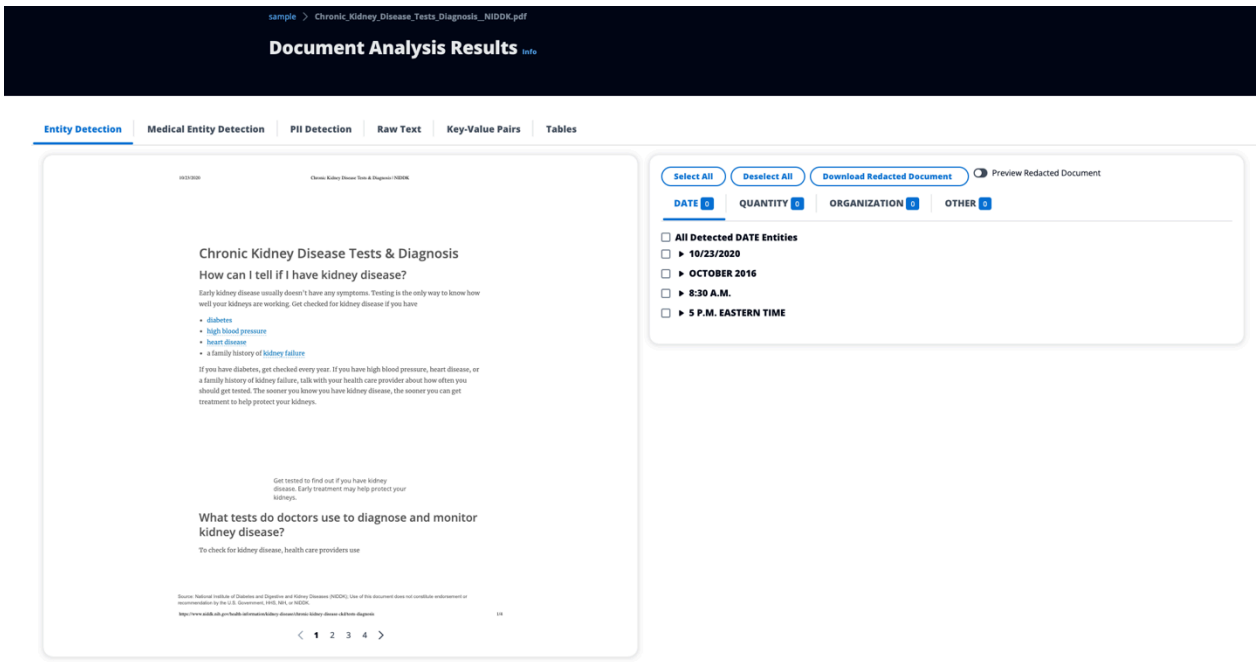


Use the analysis results components

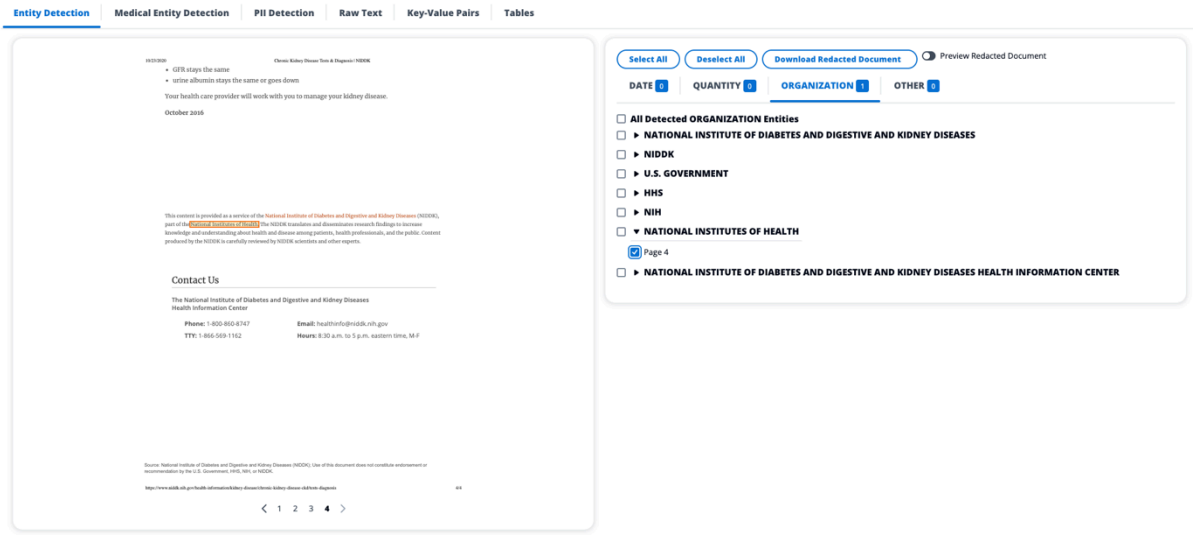
This section provides step-by-step instructions for using the components in the solution UI to see analysis results.

1. Sign in to the UI.
2. Choose a case, then choose a document.

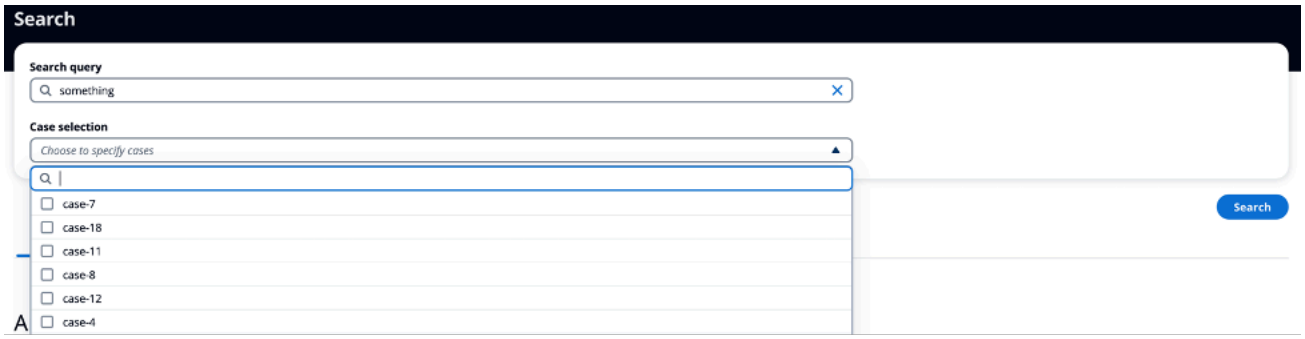




3. Select the tab for the component you want to use. The components include:
   - **Entity Detection** – Find and view entities such as dates, organizations, and quantities.
   - **Medical Entity Detection** – Find and view medical entities such as PHI and medical conditions.
   - **PII Detection** – Identify PII, which can help you identify items you need to redact.
   - **Raw Text** – View raw text elements of the document.
   - **Key-Value Pairs** – Find and view key-value pairs in the document, such as section headings and lists.
   - **Tables** – Find and view tables in the document.
4. For the **Entity Detection**, **Medical Entity Detection**, and **PII Detection** components, explore the findings by performing these actions:
   - Expand the entity to see the page(s) on which the entity exists.
   - Choose the page number to preview to the selected page.
   - Choose the entity to highlight that entity within the document.
   - Choose **Preview Redacted Document**, then choose the entity to visualize the redaction.
   - Choose **Download Redacted Document** to download the redacted version of the document.



5. For the **Raw Text**, **Key-Value Pairs**, and **Tables** components, expand the page number to see the findings.
6. If you deployed Amazon Kendra or Amazon OpenSearch Serverless during deployment, you can search for documents using the search bar.



You can filter the cases within which you would like to perform the search.

Developer guide

This section provides the source code for the solution, additional customizations, an API reference, and information for deploying the solution without the UI.

Source code

Visit our GitHub repository to download the source files for this solution and to share your customizations with others.

The Enhanced Document Understanding on AWS templates are generated using the AWS CDK. See the README.md file for additional information.

Customization guide

Refer the README.md file for information about the code and how to deploy it using CDK Toolkit.

API reference

This section provides API references for the solution.

API	HTTP Method	Functionality	Authorized Callers
`/case`	POST	Creates a new case	Cognito authenticated JWT token
`/case/{caseId}`	GET	Retrieves case details, including a list of uploaded documents	Cognito authenticated JWT token
`/cases`	GET	Retrieves a list of cases	Cognito authenticated JWT token
`/document`	POST	Uploads a new document to a case	Cognito authenticated JWT token

API	HTTP Method	Functionality	Authorized Callers
/document/download	GET	Downloads an existing document for a case	Cognito authenticated JWT token
/document/{caseId}/{documentId}	GET	Gets the S3 location with key prefix where the document is uploaded	Cognito authenticated JWT token
/inferences/{caseId}/{documentId}	GET	Gets inferences for a document associated with a case	Cognito authenticated JWT token
/inferences/{caseId}/{documentId}/{inferenceType}	GET	Get the available inferences for the document	Cognito authenticated JWT token
/redact/{caseId}/{documentId}	POST	Requests redaction of selected content in a document	Cognito authenticated JWT token
/search/kendra/{query}	GET	Searches for documents using NLP queries and Amazon Kendra service	Cognito authenticated JWT token
/search/opensearch/{query}	GET	Searches for documents using keyword queries and Amazon OpenSearch Serverless service	Cognito authenticated JWT token

Deploying the application without the UI

The solution provides an option to deploy only the API endpoints with the workflow configuration and integrate it with another UI application. Check the **Mappings** section in the CloudFormation template to enable or disable features for deployment.

Reference

This section includes information about an optional feature for collecting unique metrics for this solution and a list of builders who contributed to this solution.

Anonymized data collection

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When invoked, the following information is collected and sent to AWS:

- **Solution ID** - The AWS solution identifier
- **Unique ID (UUID)** - Randomly generated, unique identifier for each Enhanced Document Understanding on AWS deployment
- **Timestamp** - Data-collection timestamp
- **DeployAmazon KendraIndex** – Whether the solution was deployed to create an Amazon Kendra index
- **DeployOpenSearch**– Whether the solution was deployed to create an Amazon OpenSearch Serverless collection
- **WorkflowConfigName** – The name of the configuration that was deployed

AWS owns the data gathered though this survey. Data collection is subject to the AWS Privacy Notice. To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the `DocUnderstanding.template` the section called “AWS CloudFormation template” to your local hard drive.
2. Open the CloudFormation template with a text editor.
3. Modify the CloudFormation template mapping section from:

```
Mappings:
    Metrics:
      SendAnonymousUsage: "true"
```

   to:

```
Mappings:
    Metrics:
      SendAnonymousUsage: "false"
```

4. Sign in to the AWS CloudFormation console.
5. Select **Create stack**.
6. On the **Create stack** page, **Specify template** section, select **Upload a template file**.
7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in Launch the stack in the Deploy the solution section of this guide.

Contributors

- Ibrahim Mohamed
- James Nixon
- Johny Duval
- Mukit Bin Momin
- Nihit Kasabwala
- Omar Radwan
- Rad Manaktala
- Reet Takkar
- Tarek Abdunabi

Revisions

Publication date: *August 2023*

Check the CHANGELOG.md file in the GitHub repository to see all notable changes and updates to the software. The changelog provides a clear record of improvements and fixes for each version.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers, or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. AWS responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.