Developer Guide

Amazon Kendra

Amazon Kendra: Developer Guide

Table of Contents

What is Amazon Kendra?

Amazon Kendra is a managed information retrieval and intelligent search service that uses natural language processing and advanced deep learning model. Unlike traditional keyword-based search, Amazon Kendra uses semantic and contextual similarity—and ranking capabilities—to decide whether a text chunk or document is relevant to a retrieval query.

With Amazon Kendra, you can create a unified search and retrieval experience by connecting multiple data repositories to an index and ingesting and crawling documents. You can use your document metadata to create a feature-rich and customized search experience for your users, helping them efficiently find the right answers to their queries.

Amazon Kendra offers a GenAI index that's highly accurate for retrieval augmented generation (RAG) as well as enterprise search on your data. You can use Kendra GenAI indices in Amazon Q Business and Amazon Bedrock knowledge bases to build generative AI applications using your proprietary data.

> **Note**
>
> You can also use Amazon Kendra's semantic search capabilities to re-rank another search service's results. See Amazon Kendra Intelligent Ranking for more details.

What is Amazon Kendra?

Querying Amazon Kendra

You can ask Amazon Kendra the following types of queries:

Factoid questions—Simple who, what, when, or where questions, such as *Where is the nearest service center to Seattle?* Factoid questions have fact-based answers that can be returned as a single word or phrase. The answer is retrieved from a FAQ or from your indexed documents.

Descriptive questions—Questions where the answer could be a sentence, passage, or an entire document. For example, *How do I connect my Echo Plus to my network?* Or, *How do I get tax benefits for lower income families?*

Keyword and natural language questions—Questions that include complex, conversational content where the meaning may not be clear. For example, *keynote address*. When Amazon Kendra encounters a word like "address", which has multiple contextual meanings, it correctly infers the meaning behind the search query and returns relevant information.

Benefits of Amazon Kendra

Amazon Kendra is highly scalable, capable of meeting performance demands, is tightly integrated with other AWS services such as Amazon Q Business, Amazon Bedrock, Amazon S3 and Amazon Lex, and offers enterprise-grade security. Some of the benefits of using Amazon Kendra include:

Simplicity—Amazon Kendra provides a console and API for managing the documents that you want to search. You can use a simple search API to integrate Amazon Kendra into your client applications, such as websites or mobile applications.

Connectivity—Amazon Kendra can connect to third-party data repositories or data sources such as Microsoft SharePoint. You can easily index and search your documents using your data source.

Accuracy—Unlike traditional search services that use keyword searches, Amazon Kendra attempts to understand the context of the question and returns the most relevant word, snippet, or document for your query. Amazon Kendra uses machine learning to improve search results over time.

Security—Amazon Kendra delivers a highly secure enterprise search experience. Your search results reflect the security model of your organization and can be filtered based on the user or group access to documents. Customers are responsible for authenticating and authorizing user access.

Amazon Kendra Editions

Amazon Kendra offers three index types: GenAI Enterprise Edition, Basic Enterprise Edition and Basic Developer Edition.

The GenAI Enterprise Edition delivers the highest accuracy by leveraging the latest information retrieval technologies and semantic models. It offers high-availability and is built for production workloads. For the best experience, we recommend using the GenAI Index.

The Basic Enterprise Edition provides semantic search capabilities and offers a high-availability service suitable for production workloads. The Basic Developer Edition also offers semantic search capabilities, designed for building proof-of-concept solutions, but it is not recommended for production workloads.

For an overview of these indices, see Index types.

> **Note**
>
> For a list of regions, endpoints, and service quotas supported by Amazon Kendra, see Amazon Kendra endpoints and quotas.

Pricing for Amazon Kendra

You can get started for free with the Amazon Kendra GenAI Enterprise Edition index or the Amazon Kendra Developer Edition index that provides usage of up to 750 hours for the first 30 days.

> **Note**
>
> Connector usage doesn't qualify for free usage.

After your trial expires, you are charged for all provisioned Amazon Kendra indices, even if they are empty and no queries are run. After the trial expires, there are additional charges for scanning and syncing documents using the Amazon Kendra data sources.

For a complete list of charges and prices, see Amazon Kendra pricing.

Are you a first-time Amazon Kendra user?

If you are a first-time user of Amazon Kendra, we recommend that you read the following sections in order:

1 How Amazon Kendra works	2 Getting started	3 Creating an index	4 Adding documents directly to an index with batch upload	5 Creating a data source connector	6 Searching an index
Introduces Amazon Kendra components and describes how you use them to create a search solution.	Explains how to set up your account and test the Amazon Kendra search API.	Explains how to use Amazon Kendra to create a search index and to add data sources to sync your documents.	Explains how to add documents directly to an Amazon Kendra index.	Explains how to add documents from your data repository to an Amazon Kendra index.	Explains how to use the Amazon Kendra search API to search an index.

How Amazon Kendra works

Amazon Kendra provides search and Retrieval Augmented Generation (RAG) functionality for your application. It indexes your documents directly—or from your third-party document repository—and intelligently serves relevant information to your users. You can use Amazon Kendra to create an updatable index of documents of a variety of types. For a list of document types supported by Amazon Kendra, see Types of documents.

Amazon Kendra integrates with other services. You can connect an Amazon Kendra GenAI Enterprise Edition index to Amazon Q Business and Amazon Bedrock for creating your RAG chat solution. Or, you can power Amazon Lex chat bots with Amazon Kendra search to provide useful answers to users' questions. You can also use an Amazon Simple Storage Service bucket as a data source for Amazon Kendra to connect to and index your documents.

Amazon Kendra has the following components:

- An *index* that holds your documents and makes them searchable.
- A *data source* that stores your documents and Amazon Kendra connects to. You can automatically synchronize a data source with an Amazon Kendra index so that your index stays updated with your source repository.
- A *document addition API* that adds documents directly to an index.
- A *retrieve API* that retrieves relevant passages or text excerpts given an input query.
- A *query API* that searches an index given an input query.

You can use Amazon Kendra through the console or the API. You can create, update, and delete indexes. Deleting an index deletes all of its data source connectors and permanently deletes all of your document information from Amazon Kendra.

Topics

Indexes in Amazon Kendra

An index holds the contents of your documents and is structured in a way to make the documents searchable. This section provides an overview of the supported index types and index features.

Topics

- Index types in Amazon Kendra
- Adding documents to an index in Amazon Kendra
- Using Amazon Kendra reserved or common document fields
- Retrieving responses from indexes in Amazon Kendra

Index types in Amazon Kendra

Amazon Kendra has three index types: GenAI Enterprise Edition index, Enterprise Edition index, and Developer Edition index. The following sections describe the features of each index.

Topics

- Amazon Kendra GenAI Enterprise Edition index
- Amazon Kendra Enterprise Edition index
- Amazon Kendra Developer Edition index

Amazon Kendra GenAI Enterprise Edition index

An Amazon Kendra GenAI Enterprise Edition index offers the highest accuracy for the Retrieve API operation and for Retrieval Augmented Generation (RAG) use cases. It's powered by the latest information retrieval technologies—like hybrid search (keyword and vector), semantic embedding, and re-ranker models—and has been tested across a variety of datasets. The Query API operation offers similar accuracy for an Amazon Kendra GenAI Enterprise Edition index when compared with Amazon Kendra Developer Edition and Amazon Kendra Enterprise Edition indexes.

An Amazon Kendra GenAI Enterprise Edition index enables mobility of your indexed data across AWS generative AI services. With this functionality, you can seamlessly reuse your investments without having to rebuild indexes. You can use it in an Amazon Bedrock knowledge base as a managed retriever, and integrate it with Amazon Bedrock tools like agents and prompt flows to build advanced AI assistants. You can also use it with Amazon Q Business for a fully managed digital assistant.

An Amazon Kendra GenAI Enterprise Edition index offers smaller, more granular capacity units and a lower starting price compared to the other two index types. This helps you to be more efficient with your capacity utilization.

> **Note**
>
> For the best experience and accuracy, we recommend that you choose an Amazon Kendra GenAI Enterprise Edition index.

Topics

Supported features

The following features are supported for an Amazon Kendra GenAI Enterprise Edition index if you're using the Retrieve API operation for RAG use cases:

- **Full support** – Confidence score buckets, filtering, faceting, relevance tuning, custom document enrichment, custom metadata, and adjusting query capacity and document capacity.
- **Partial support** – Data source connectors and user context filtering. For more information on partially supported features, see Limitations.

The following features are supported for an Amazon Kendra GenAI Enterprise Edition index if you're using the Query API operation for search use cases:

- **Full support** – Document ranking, extractive question answering, confidence score buckets, filtering, faceting, sorting, collapsing and expanding query results, index browsing, Boolean queries, exact match, wildcard queries, query suggestions, query spell checker, relevance tuning, incremental learning, custom document enrichment, custom metadata, adjusting query capacity and document capacity, and search experience.
- **Partial support** – Data source connectors and user context filtering. For more information on partially supported features, see Limitations.

Limitations

The following outlines the known limitations of an Amazon Kendra GenAI Enterprise Edition index:

- Amazon Kendra GenAI Enterprise Edition indexes are only available in US East (N. Virginia) and US West (Oregon).
- Amazon Kendra GenAI Enterprise Edition indexes only support English language content.
- Amazon Kendra GenAI Enterprise Edition indexes support only v2.0 Amazon Kendra data source connectors.
- In an Amazon Kendra GenAI Enterprise Edition index, you can only use user attributes to filter search results by user context.
- Amazon Kendra GenAI Enterprise Edition indexes don't support token-based user access control or user ID and group –based user access control to documents.
- The CreateAccessControlConfiguration API operation is disabled for Amazon Kendra GenAI Enterprise Edition indexes.
- If you're using an Amazon Kendra GenAI Enterprise Edition index with Amazon Q Business, note the following about controlling end-user access to documents:

  Amazon Q Business uses user email ID to determine end-user access to documents in an index. When you connect an Amazon Kendra index to Amazon Q Business, Amazon Q Business relays the user's identifying email ID to Amazon Kendra to enable document filtering for end users. If data sources connected to your Amazon Kendra index don't use email ID–based document filtering, or the email ID is not present, Amazon Q Business generates responses only from public documents.

Amazon Kendra Enterprise Edition index

An Amazon Kendra Enterprise Edition index provides semantic search capabilities, and offers a high-availability service that is suitable for production workloads.

Topics

Supported features

The following features are supported for an Amazon Kendra Enterprise Edition index if you're using the Retrieve API operation for RAG use cases: querying using advance query syntax, suggested spell corrections for queries, faceting, query suggestions to autocomplete search queries, and incremental learning.

All features are supported for an Amazon Kendra Enterprise Edition index if you're using the Query API operation for search use cases.

Limitations

The following outlines the known limitations of an Amazon Kendra Enterprise Edition index:

- If you're using an Amazon Kendra Enterprise Edition index with Amazon Q Business, note the following about controlling end-user access to documents:

  Amazon Q Business uses user email ID to determine end-user access to documents in an index. When you connect an Amazon Kendra index to Amazon Q Business, Amazon Q Business relays the user’s identifying email ID to Amazon Kendra to enable document filtering for end users. If data sources connected to your Amazon Kendra index don’t use email ID–based document filtering, or the email ID is not present, Amazon Q Business generates responses only from public documents.

Amazon Kendra Developer Edition index

An Amazon Kendra Developer Edition index also provides semantic search capabilities for you to test your use cases. However, we don't recommend it for production use cases.

Topics

Supported features

The following features are supported for an Amazon Kendra Developer Edition index if you're using the Retrieve API operation for RAG use cases: querying using advance query syntax, suggested spell corrections for queries, faceting, query suggestions to autocomplete search queries, and incremental learning.

All features are supported for an Amazon Kendra Developer Edition index if you're using the Query API operation for search use cases.

Limitations

The following outlines the known limitations of an Amazon Kendra Developer Edition index:

- If you're using an Amazon Kendra Developer Edition index with Amazon Q Business, note the following about controlling end-user access to documents:

  Amazon Q Business uses user email ID to determine end-user access to documents in an index. When you connect an Amazon Kendra index to Amazon Q Business, Amazon Q Business relays the user's identifying email ID to Amazon Kendra to enable document filtering for end users. If data sources connected to your Amazon Kendra index don't use email ID–based document filtering, or the email ID is not present, Amazon Q Business generates responses only from public documents.

Adding documents to an index in Amazon Kendra

The way you add documents to an index depends on how you store your documents.

- If you store your documents in some kind of repository, such as an Amazon S3 bucket or a Microsoft SharePoint site, you use a data source connector to index your documents from your repository.
- If you don't store your documents in a repository, you use the BatchPutDocument API operation to directly index your documents.
- For FAQ questions and answers, which must be stored in an Amazon Kendra (Amazon S3) bucket, you upload them from the bucket.

You can create indexes with the Amazon Kendra console, the AWS CLI, or an AWS SDK. For information about the types of documents that can be indexed, see Document types.

Using Amazon Kendra reserved or common document fields

With the UpdateIndex API operation, you can create reserved or common fields. To do this, you use `DocumentMetadataConfigurationUpdates`, and specify the Amazon Kendra reserved index field name to map to your equivalent document attribute/field name. You can also create custom fields.

If you use a data source connector, most include field mappings that map your data source document fields to Amazon Kendra index fields. If you use the console, you update fields by selecting your data source, selecting the edit action, and then proceeding next to the field mappings section for configuring the data source.

You can configure the `Search` object to set a field as either displayable, facetable, searchable, and sortable. You can configure the `Relevance` object to set a field's rank order, boost duration or time period to apply to boosting, freshness, importance value, and importance values mapped to specific field values.

If you use the console, you can configure the search settings for a field by selecting the facet option in the navigation menu. To set relevance tuning, select the option to search your index in the navigation menu, enter a query, and use the side panel options to tune the search relevance. You can't change the field type after you have created the field.

Amazon Kendra has the following reserved or common document fields that you can use:

- `_authors` – A list of one or more authors responsible for the content of the document.
- `_category` – A category that places a document in a specific group.
- `_created_at` – The date and time in ISO 8601 format that the document was created. For example, 2012-03-25T12:30:10+01:00 is the ISO 8601 date-time format for March 25th, 2012 at 12:30 PM (plus 10 seconds) in Central European Time.
- `_data_source_id` – The identifier of the data source that contains the document.
- `_document_body` – The content of the document.
- `_document_id` – A unique identifier for the document.
- `_document_title` – The title of the document.
- `_excerpt_page_number` – The page number in a PDF file where the document excerpt appears. If your index was created before September 8, 2020, you must re-index your documents before you can use this attribute.
- `_faq_id` – If this is a question-answer type document (FAQ), a unique identifier for the FAQ.
- `_file_type` – The file type of the document, such as pdf or doc.
- `_last_updated_at` – The date and time in ISO 8601 format that the document was last updated. For example, 2012-03-25T12:30:10+01:00 is the ISO 8601 date-time format for March 25th, 2012 at 12:30 PM (plus 10 seconds) in Central European Time.
- `_source_uri` – The URI where the document is available—for example, the URI of the document on a company website.

- `_version` – An identifier for the specific version of a document.
- `_view_count` – The number of times that the document has been viewed.
- `_language_code` (String) – The code for a language that applies to the document. This defaults to English if you don't specify a language. For more information on supported languages, including their codes, see Adding documents in languages other than English.

You create custom fields using `DocumentMetadataConfigurationUpdates` with the `UpdateIndex` API operation, just as you do when creating a reserved or common field. You must set the appropriate data type for your custom field.

If you use the console, you update fields by selecting your data source, selecting the edit action, and then proceeding next to the field mappings section for configuring the data source. Some data sources don't support adding new fields or custom fields. You can't change the field type after you have created the field.

The following are the types that you can set for custom fields:

- Date
- Number
- String
- String list

If you added documents to an index using the BatchPutDocument API operation, `Attributes` lists the fields/attributes of your documents and you create fields using the `DocumentAttribute` object.

For documents indexed from an Amazon S3 data source, you create fields using a JSON metadata file that includes the fields information.

If you use a supported database as your data source, you can configure your fields using the field mappings option.

Retrieving responses from indexes in Amazon Kendra

After you create an index, you can start searching your documents.

To search an Amazon Kendra index, you use either the Retrieve API operation or the Query API operation.

The Retrieve API operation is ideal for Retrieval Augmented Generation (RAG) use cases. For a given query, it returns a ranked list of semantically relevant passages of up to 200 token words. You can send these to a large language model (LLM) to generate an answer using RAG. For more information, see Searching an index.

The Query API operation is best for document search use cases. For a given query, it returns a list of ranked documents with 100 word excerpts that are relevant to the query. This is useful for traditional document search use cases where users are browsing through a list of ranked documents.

To see what features are supported by the Retrieve and Query API operations for each index type, see Index types.

Documents

This section explains how Amazon Kendra indexes the many document formats it supports and the different fields/attributes of documents.

Topics

- Document types or formats
- Document attributes or fields

Document types or formats

Amazon Kendra supports popular document types or formats such as PDF, HTML, Word, PowerPoint, and more. An index can contain multiple document formats.

Amazon Kendra extracts the content inside the documents in order to make the documents searchable. The documents are parsed in a way to optimize search on the extracted text and any tabular content (HTML tables) within the documents. This means structuring the documents into fields or attributes that are used for search. The document metadata, such as the last modified date, can be useful fields for search.

Documents can be organized into rows and columns. For example, each document is a row and each document field/attribute, such as the title and body content, is a column. For example, if you use a database as your data source, the data should be structured or organized into rows and columns.

You can add documents to your index through the following ways:

- BatchPutDocument API
- Data source connector

If you want to add a FAQ file, you use the CreateFaq API to add the file stored in an Amazon S3 bucket. You can choose between a basic CSV format, a CSV format that includes custom fields/ attributes in a header, and a JSON format that includes custom fields. The default format is basic CSV.

The following provides information on each supported document format and how Amazon Kendra treats each format when indexing documents.

Document format	Treated as	How document is treated	Original structure
Portable Document Format (PDF)	HTML	Converted to HTML, then content is extracted.	Unstructured
HyperText Markup Language (HTML)	HTML	HTML tags are filtered out to extract content. Content must between the main HTML start and closing tags (`<HTML>content</HTML>`).	Semi-structured
Extensible Markup Language (XML)	XML	XML tags are filtered out to extract content.	Semi-structured
Extensible Stylesheet Language Transformation (XSLT)	XSLT	Tags are filtered out to extract content.	Semi-structured

Document format	Treated as	How document is treated	Original structure
MarkDown (MD)	Plain text	Content is extracted with MarkDown syntax included.	Semi-structured
Comma Separated Values (CSV)	CSV	Content extracted from each cell, with a single file treated as a single document result.	Structured for FAQ files, otherwise semi-structured
Microsoft Excel (XLS and XLSX)	XLS and XLSX	Content extracted from each cell, with a single file treated as a single document result.	Semi-structured
JavaScript Object Notation (JSON)	Plain text	Content is extracted with JSON syntax included.	Semi-structured
Rich Text Format (RTF)	RTF	RTF syntax is filtered out to extract content.	Semi-structured
Microsoft PowerPoint (PPT)	PPT, PPTX	Only text content is extracted from PowerPoint slides for search. Images and other content are not extracted.	Unstructured

Document format	Treated as	How document is treated	Original structure
Microsoft Word	DOC, DOCX	Only text content is extracted from Word pages for search. Images and other content are not extracted.	Unstructured
Plain text (TXT)	TXT	All text in the text document is extracted.	Unstructured

Document attributes or fields

A document has attributes or fields associated with it. Fields of a document are the properties of a document or what is contained within the structure of a document. For example, each of your documents might contain title, body text, and author. You can also add custom fields for your particular documents. For example, if your index searches tax documents, you might specify a custom field for the type of tax document such as W-2, 1099, and so on.

Before you can use a document field in a query, it must be mapped to an index field. For example, the title field can be mapped to the field `_document_title`. For more information, see Mapping fields. To add a new field, you must create an index field to map the field to. You create index fields using the console or by using the UpdateIndex API.

You can use document fields to filter responses and to make faceted search results. For example, you can filter a response to only return a specific version of a document, or you can filter searches to only return 1099 type of tax documents that match the search term. For more information, see Filtering and facet search.

You can also use document fields to manually tune the query response. For example, you can choose to increase the importance of the title field to increase the weight that Amazon Kendra assigns to the field when determining which documents to return in the response. For more information, see Tuning search relevance.

If you are adding a document directly to an index, you specify the fields in the Document input parameter to the BatchPutDocument API. You specify the custom field values in a DocumentAttribute object array. If you are using a data source, the method that you use to add the document fields depends on the data source. For more information, see Mapping data source fields.

Using Amazon Kendra reserved or common document fields

With the UpdateIndex API, you can create reserved or common fields using `DocumentMetadataConfigurationUpdates` and specifying the Amazon Kendra reserved index field name to map to your equivalent document attribute/field name. You can also create custom fields. If you use a data source connector, most include field mappings that map your data source document fields to Amazon Kendra index fields. If you use the console, you update fields by selecting your data source, selecting the edit action, and then proceeding next to the field mappings section for configuring the data source.

You can configure the `Search` object to set a field as either displayable, facetable, searchable, and sortable. You can configure the `Relevance` object to set a field's rank order, boost duration or time period to apply to boosting, freshness, importance value, and importance values mapped to specific field values. If you use the console, you can set the search settings for a field by selecting the facet option in the navigation menu. To set relevance tuning, select the option to search your index in the navigation menu, enter a query, and use the side panel options to tune the search relevance. You cannot change the field type once you have created the field.

Amazon Kendra has the following reserved or common document fields that you can use:

- `_authors`—A list of one or more authors responsible for the content of the document.
- `_category`—A category that places a document in a specific group.
- `_created_at`—The date and time in ISO 8601 format that the document was created. For example, 2012-03-25T12:30:10+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.
- `_data_source_id`—The identifier of the data source that contains the document.
- `_document_body`—The content of the document.
- `_document_id`—A unique identifier for the document.
- `_document_title`—The title of the document.

- `_excerpt_page_number`—The page number in a PDF file where the document excerpt appears. If your index was created before September 8, 2020, you must re-index your documents before you can use this attribute.
- `_faq_id`—If this is a question-answer type document (FAQ), a unique identifier for the FAQ.
- `_file_type`—The file type of the document, such as pdf or doc.
- `_last_updated_at`—The date and time in ISO 8601 format that the document was last updated. For example, 2012-03-25T12:30:10+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.
- `_source_uri`—The URI where the document is available. For example, the URI of the document on a company website.
- `_version`—An identifier for the specific version of a document.
- `_view_count`—The number of times that the document has been viewed.
- `_language_code` (String)—The code for a language that applies to the document. This defaults to English if you do not specify a language. For more information on supported languages, including their codes, see [Adding documents in languages other than English].

For custom fields, you create these fields using `DocumentMetadataConfigurationUpdates` with the `UpdateIndex` API, just as you do when creating a reserved or common field. You must set the appropriate data type for your custom field. If you use the console, you update fields by selecting your data source, selecting the edit action, and then proceeding next to the field mappings section for configuring the data source. Some data sources don't support adding new fields or custom fields. You cannot change the field type once you have created the field.

The following are the types you can set for custom fields:

- Date
- Number
- String
- String list

If you added documents to the index using [BatchPutDocument] API, `Attributes` lists the fields/attributes of your documents and you create fields using the `DocumentAttribute` object.

For documents indexed from an Amazon S3 data source, you create fields using a [JSON metadata file] that includes the fields information.

If you use a supported database as your data source, you can configure your fields using the field mappings option.

Data sources

A data source is a data repository or location that Amazon Kendra connects to and indexes your documents or content. For example, you can configure Amazon Kendra to connect to Microsoft SharePoint to crawl and index your documents stored in this source. You can also index web pages by providing the URLs for Amazon Kendra to crawl. You can automatically synchronize a data source with an Amazon Kendra index so that added, updated, or deleted documents in the data source are also added, updated, or deleted in the index.

Supported data sources are:

- Adobe Experience Manager
- Alfresco
- Aurora (MySQL)
- Aurora (PostgreSQL)
- Amazon FSx (Windows)
- Amazon FSx (NetApp ONTAP)
- Database data sources
- Amazon RDS (Microsoft SQL Server)
- Amazon RDS (MySQL)
- Amazon RDS (Oracle)
- Amazon RDS (PostgreSQL)
- Amazon S3 buckets
- Amazon Kendra Web Crawler
- Box
- Confluence
- Custom data sources
- Dropbox
- Drupal

- GitHub
- Gmail
- Google Workspace Drives
- IBM DB2
- Jira
- Microsoft Exchange
- Microsoft OneDrive
- Microsoft SharePoint
- Microsoft Teams
- Microsoft SQL Server
- Microsoft Yammer
- MySQL
- Oracle Database
- PostgreSQL
- Quip
- Salesforce
- ServiceNow
- Slack
- Zendesk

For a list of document types or formats supported by Amazon Kendra see Document types. You must first create an index before creating a data source connector to index your documents from your data source.

> **Note**
>
> To create an index of documents, you don't need to use a data source. You can add documents directly to an index with batch upload. For more information, see Adding documents directly to an index.

For a walkthrough on using the Amazon Kendra console, the AWS CLI, or SDKs, see Getting started.

Queries

To get answers, users query an index. Users can use natural language in their queries. The response contains information, such as the title, a text excerpt, and the location of documents in the index that provide the best answer.

Amazon Kendra uses all of the information that you provide about your documents, not just the contents of the documents, to determine whether a document is relevant to the query. For example, if your index contains information about when documents were last updated, you can tell Amazon Kendra to assign a higher relevance to documents that were updated more recently.

A query can also contain criteria for how to filter the response so that Amazon Kendra returns only documents that satisfy the filter criteria. For example, if you created an index field called *department*, you can filter the response so that only documents with the department field set to *legal* are returned. For more information, see Filtering search.

You can influence the results of a query by tuning the relevance of individual fields in the index. Tuning changes the importance of a field on the results. For example, if you raise the importance of documents with the category *new*, documents with this category are more likely to be included in the response. For more information, see Tuning search relevance.

For more information about using queries, see Searching an index.

Tags

Manage your indices, data sources, and FAQs by assigning tags or labels. You can use tags to categorize your Amazon Kendra resources in various ways. For example, by purpose, owner, or application, or any combination. Each tag consists of a *key* and a *value*, both of which you define.

Tags help you to:

- Identify and organize your AWS resources. Many AWS services support tagging, so you can assign the same tag to resources in different services to indicate that the resources are related. For example, you can tag an index and the Amazon Lex bot that uses the index with the same tag.
- Allocate costs. You activate tags on the AWS Billing and Cost Management dashboard. AWS uses tags to categorize your costs and deliver a monthly cost allocation report to you. For more information, see Cost Allocation and Tagging in *About AWS Billing and Cost Management*.
- Control access to your resources. You can use tags in AWS Identity and Access Management (IAM) policies that control access to Amazon Kendra resources. You can attach these policies to an IAM

role or user to activate tag-based access control. For more information, see Authorization based on tags.

You can create and manage tags using the AWS Management Console, the AWS Command Line Interface (AWS CLI), or the Amazon Kendra API.

Tagging resources

If you're using the Amazon Kendra console, you can tag resources when you create them or add them later. You can also use the console to update or remove tags.

If you're using the AWS Command Line Interface (AWS CLI) or the Amazon Kendra API, use the following operations to manage tags for your resources:

- CreateDataSource—Apply tags when you create a data source.
- CreateFaq—Apply tags when you create an FAQ.
- CreateIndex—Apply tags when you create an index.
- ListTagsForResource—View the tags associated with a resource.
- TagResource—Add and modify tags for a resource.
- UntagResource—Remove tags from a resource.

Tag restrictions

The following restrictions apply to tags on Amazon Kendra resources:

- Maximum number of tags—50
- Maximum key length—128 characters
- Maximum value length—256 characters
- Valid characters for key and value—a–z, A–Z, space, and the following characters: _ . : / = + - and @
- Keys and values are case sensitive
- Don't use `aws:` as a prefix for keys; it's reserved for AWS use

Setting up Amazon Kendra

Before using Amazon Kendra, you must have an Amazon Web Services (AWS) account. After you have an AWS account, you can access Amazon Kendra through the Amazon Kendra console, the AWS Command Line Interface (AWS CLI), or the AWS SDKs.

This guide includes examples for AWS CLI, Java, and Python.

Topics

Sign up for AWS

When you sign up for Amazon Web Services (AWS), your account is automatically signed up for all services in AWS, including Amazon Kendra. You are charged only for the services that you use.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To sign up for AWS

1. Open https://aws.amazon.com, and then choose **Create an AWS Account**.
2. Follow the on-screen instructions to complete the account creation. Note your 12-digit AWS account number. Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.
3. Create an AWS Identity and Access Management (IAM) admin user. See Creating Your First IAM User and Group in the *AWS Identity and Access Management User Guide* for instructions.

Regions and endpoints

An endpoint is a URL that is the entry point for a web service. Each endpoint is associated with a specific AWS region. If you use a combination of the Amazon Kendra console, the AWS CLI, and the

Amazon Kendra SDKs, pay attention to their default regions as all Amazon Kendra components of a given campaign (index, query, etc.) must be created in the same region. For the regions and endpoints supported by Amazon Kendra, see Regions and Endpoints.

Setting up the AWS CLI

The AWS Command Line Interface (AWS CLI) is a unified developer tool for managing AWS services, including Amazon Kendra. We recommend that you install it.

1. To install the AWS CLI, follow the instructions in Installing the AWS Command Line Interface in the *AWS Command Line Interface User Guide*.
2. To configure the AWS CLI and set up a profile to call the AWS CLI, follow the instructions in Configuring the AWS CLI in the *AWS Command Line Interface User Guide*.
3. To confirm that the AWS CLI profile is configured properly, run the following command:

   ```
   aws configure --profile default
   ```

   If your profile has been configured correctly, you will see output similar to the following:

   ```
   AWS Access Key ID [****************52FQ]:
   AWS Secret Access Key [****************xgyZ]:
   Default region name [us-west-2]:
   Default output format [json]:
   ```
4. To verify that the AWS CLI is configured for use with Amazon Kendra, run the following commands:

   ```
   aws kendra help
   ```

   If the AWS CLI is configured correctly, you will see a list of the supported AWS CLI commands for Amazon Kendra, Amazon Kendra runtime, and Amazon Kendra events.

Setting up the AWS SDKs

Download and install the AWS SDKs that you want to use. This guide provides examples for Python. For information about other AWS SDKs, see Tools for Amazon Web Services.

The package for the Python SDK is called *Boto3*.

Before you run the below Python commands, you must first download and install Python 3.6 or later for your operating system. Support for Python 3.5 and earlier is deprecated. If you do not have pip included in your Python Scripts directory, you can download get-pip.py and store this in your Scripts directory. You can also set your Python directory as a Path or environment variable using a terminal program.

```
# Install the latest Boto3 release via pip
pip install boto3

# You can install a specific version of Boto3 for compatibility reasons
# Install Boto3 version 1.0 specifically
pip install boto3==1.0.0

# Make sure Boto3 is no older than version 1.15.0
pip install boto3>=1.15.0

# Avoid versions of Boto3 newer than version 1.15.3
pip install boto3<=1.15.3
```

To use Boto3, you must set up authentication credentials for your AWS account using the IAM console.

IAM access roles for Amazon Kendra

When you create an index, data source, or an FAQ, Amazon Kendra needs access to the AWS resources required to create the Amazon Kendra resource. You must create a AWS Identity and Access Management (IAM) policy before you create the Amazon Kendra resource. When you call the operation, you provide the Amazon Resource Name (ARN) of the role with the policy attached. For example, if you are calling the BatchPutDocument API to add documents from an Amazon S3 bucket, you provide Amazon Kendra with a role with a policy that has access to the bucket.

You can create a new IAM role in the Amazon Kendra console or choose an IAM existing role to use. The console displays roles that have the string "kendra" or "Kendra" in the role name.

The following topics provide details for the required policies. If you create IAM roles using the Amazon Kendra console these policies are created for you.

Topics

IAM roles for indexes

When you create an index, you must provide an IAM role with permission to write to an Amazon CloudWatch. You must also provide a trust policy that allows Amazon Kendra to assume the role. The following are the policies that must be provided.

IAM roles for indexes

A role policy to allow Amazon Kendra to access a CloudWatch log.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "AWS/Kendra"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "logs:DescribeLogGroups",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
kendra/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogStreams",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
kendra/*:log-stream:*"
        }
    ]
}
```

A role policy to allow Amazon Kendra to access AWS Secrets Manager. If you are using user context with Secrets Manager as a key location, you can use the following policy.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "AWS/Kendra"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "logs:DescribeLogGroups",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
kendra/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogStreams",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
kendra/*:log-stream:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
```

```
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for the BatchPutDocument API

> **Warning**
>
> Amazon Kendra doesn't use a bucket policy that grants permissions to an Amazon Kendra principal to interact with an S3 bucket. Instead, it uses IAM roles. Make sure that Amazon Kendra isn't included as a trusted member in your bucket policy to avoid any data security issues in accidentally granting permissions to arbitrary principals. However, you can add a bucket policy to use an Amazon S3 bucket across different accounts. For more information, see Policies to use Amazon S3 across accounts. For information about IAM roles for S3 data sources, see IAM roles.

When you use the BatchPutDocument API to index documents in an Amazon S3 bucket, you must provide Amazon Kendra with an IAM role with access to the bucket. You must also provide a trust policy that allows Amazon Kendra to assume the role. If the documents in the bucket are encrypted, you must provide permission to use the AWS KMS customer master key (CMK) to decrypt the documents.

IAM roles for the BatchPutDocument API

A required role policy to allow Amazon Kendra to access an Amazon S3 bucket.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name/*"
            ]
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

It is recommended that you include `aws:sourceAccount` and `aws:sourceArn` in the trust policy. This limits permissions and securely checks if `aws:sourceAccount` and `aws:sourceArn` are the same as provided in the IAM role policy for the `sts:AssumeRole` action. This prevents unauthorized entities from accessing your IAM roles and their permissions. For more information, see the AWS Identity and Access Management guide on the confused deputy problem.

An optional role policy to allow Amazon Kendra to use an AWS KMS customer master key (CMK) to decrypt documents in an Amazon S3 bucket.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ]
```

```
        }
    ]
}
```

IAM roles for data sources

When you use the CreateDataSource API, you must give Amazon Kendra an IAM role that has permission to access the resources. The specific permissions required depend on the data source.

IAM roles for Adobe Experience Manager data sources

When you use Adobe Experience Manager, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Adobe Experience Manager.
- Permission to call the required public APIs for the Adobe Experience Manager connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect an Adobe Experience Manager data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret-id"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:PutPrincipalMapping",
                "kendra:DeletePrincipalMapping",
                "kendra:ListGroupsOlderThanOrderingId",
                "kendra:DescribePrincipalMapping"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:111122223333:index/index-id",
                "arn:aws:kendra:us-east-1:111122223333:index/index-id/data-
source/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:111122223333:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Alfresco data sources

When you use Alfresco, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Alfresco.
- Permission to call the required public APIs for the Alfresco connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect an Alfresco data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:PutPrincipalMapping",
                "kendra:DeletePrincipalMapping",
                "kendra:ListGroupsOlderThanOrderingId",
                "kendra:DescribePrincipalMapping"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:123456789012:index/index-id",
                "arn:aws:kendra:us-east-1:123456789012:index/index-id/data-
source/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
```

```
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Aurora (MySQL) data sources

When you use Aurora (MySQL), you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Aurora (MySQL).
- Permission to call the required public APIs for the Aurora (MySQL) connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect an Aurora (MySQL) data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Principal":{
                "Service":"kendra.amazonaws.com"
            },
            "Action":"sts:AssumeRole"
        }
    ]
}
```

IAM roles for Aurora (PostgreSQL) data sources

When you use Aurora (PostgreSQL), you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Aurora (PostgreSQL).
- Permission to call the required public APIs for the Aurora (PostgreSQL) connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect an Aurora (PostgreSQL) data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Amazon FSx data sources

When you use Amazon FSx, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Amazon FSx file system.
- Permission to access Amazon Virtual Private Cloud (VPC) where your Amazon FSx file system resides.
- Permission to get the domain name of your Active Directory for your Amazon FSx file system.
- Permission to call the required public APIs for the Amazon FSx connector.
- Permission to call the `BatchPutDocument` and `BatchDeleteDocument` APIs to update the index.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for database data sources

When you use a database as a data source, you provide Amazon Kendra with a role that has the permissions necessary for connecting to the . These include:

- Permission to access the AWS Secrets Manager secret that contains the user name and password for the site. For more information about the contents of the secret, see data sources.
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by Secrets Manager.
- Permission to use the `BatchPutDocument` and `BatchDeleteDocument` operations to update the index.
- Permission to access the Amazon S3 bucket that contains the SSL certificate used to communicate with the site.

> **Note**
>
> You can connect database data sources to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:your-region:your-account-id:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:your-region:your-account-id:key/key-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": [
                "arn:aws:kendra:your-region:your-account-id:index/index-id"
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "kendra.your-region.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::bucket-name/*"
            ]
        }
    ]
}
```

There are two optional policies that you might use with a data source.

If you have encrypted the Amazon S3 bucket that contains the SSL certificate used to communicate with the , provide a policy to give Amazon Kendra access to the key.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ]
        }
    ]
}
```

If you are using a VPC, provide a policy that gives Amazon Kendra access to the required resources. See IAM roles for data sources, VPC for the required policy.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
```

```
            "Principal":{
               "Service":"kendra.amazonaws.com"
            },
            "Action":"sts:AssumeRole"
         }
      ]
   }
```

IAM roles for Amazon RDS (Microsoft SQL Server) data sources

When you use a Amazon RDS (Microsoft SQL Server) data source connector, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Amazon RDS (Microsoft SQL Server) data source instance.
- Permission to call the required public APIs for the Amazon RDS (Microsoft SQL Server) data source connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect an Amazon RDS (Microsoft SQL Server) data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
```

```
            "Principal":{
               "Service":"kendra.amazonaws.com"
            },
            "Action":"sts:AssumeRole"
         }
      ]
}
```

IAM roles for Amazon RDS (MySQL) data sources

When you use a Amazon RDS (MySQL) data source connector, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Amazon RDS (MySQL) data source instance.
- Permission to call the required public APIs for the Amazon RDS (MySQL) data source connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect an Amazon RDS (MySQL) data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
```

```
            },
            "Action":"sts:AssumeRole"
        }
    ]
}
```

IAM roles for Amazon RDS (Oracle) data sources

When you use a Amazon RDS Oracle data source connector, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Amazon RDS (Oracle) data source instance.
- Permission to call the required public APIs for the Amazon RDS (Oracle) data source connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

Note

You can connect an Amazon RDS Oracle data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Principal":{
                "Service":"kendra.amazonaws.com"
            },
            "Action":"sts:AssumeRole"
```

```
        }
    ]
}
```

IAM roles for Amazon RDS (PostgreSQL) data sources

When you use a Amazon RDS (PostgreSQL) data source connector, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Amazon RDS (PostgreSQL) data source instance.
- Permission to call the required public APIs for the Amazon RDS (PostgreSQL) data source connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect an Amazon RDS (PostgreSQL) data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Principal":{
                "Service":"kendra.amazonaws.com"
            },
            "Action":"sts:AssumeRole"
        }
```

```
    ]
}
```

IAM roles for Amazon S3 data sources

> **Warning**
>
> Amazon Kendra doesn't use a bucket policy that grants permissions to an Amazon Kendra principal to interact with an S3 bucket. Instead, it uses IAM roles. Make sure that Amazon Kendra isn't included as a trusted member in your bucket policy to avoid any data security issues in accidentally granting permissions to arbitrary principals. However, you can add a bucket policy to use an Amazon S3 bucket across different accounts. For more information, see Policies to use Amazon S3 across accounts (scroll down).

When you use an Amazon S3 bucket as a data source, you supply a role that has permission to access the bucket, and to use the `BatchPutDocument` and `BatchDeleteDocument` operations. If the documents in the Amazon S3 bucket are encrypted, you must provide permission to use the AWS KMS customer master key (CMK) to decrypt the documents.

The following role policies must allow Amazon Kendra to assume a role. Scroll further down to view a trust policy to assume a role.

A required role policy to allow Amazon Kendra to use an Amazon S3 bucket as a data source.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name/*"
            ],
            "Effect": "Allow"
        },
```

```
        {
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name"
            ],
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:123456789012:index/index-id"
            ]
        }
    ]
}
```

An optional role policy to allow Amazon Kendra to use an AWS KMS customer master key (CMK) to decrypt documents in an Amazon S3 bucket.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ]
        }
    ]
}
```

An optional role policy to allow Amazon Kendra to access an Amazon S3 bucket, while using a Amazon VPC, and without activating AWS KMS or sharing AWS KMS permissions.

An optional role policy to allow Amazon Kendra to access an Amazon S3 bucket while using a Amazon VPC, and with AWS KMS permissions activated.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Policies to use Amazon S3 across accounts

If your Amazon S3 bucket is in a different account to the account you use for your Amazon Kendra index, you can create policies to use it across accounts.

A role policy to use your Amazon S3 bucket as your data source when the bucket is in a different account to your Amazon Kendra index. Note that `s3:PutObject` and `s3:PutObjectAcl` are optional, and you use this if you want to include a configuration file for your access control list.

A bucket policy to allow the Amazon S3 data source role to access the Amazon S3 bucket across accounts. Note that `s3:PutObject` and `s3:PutObjectAcl` are optional, and you use this if you want to include a configuration file for your access control list.

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "$kendra-s3-connector-role-arn"
            },
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::$bucket-in-other-account/*"
            ]
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "$kendra-s3-connector-role-arn"
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::$bucket-in-other-account"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
```

```
    ]
}
```

IAM roles for Amazon Kendra Web Crawler data sources

When you use Amazon Kendra Web Crawler, you provide a role with the following policies:

- Permission to access the AWS Secrets Manager secret that contains the credentials to connect to websites or a web proxy server backed by basic authentication. For more information about the contents of the secret, see Using a web crawler data source.
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by Secrets Manager.
- Permission to use the `BatchPutDocument` and `BatchDeleteDocument` operations to update the index.
- If you use an Amazon S3 bucket to store your list of seed URLs or sitemaps, include permission to access the Amazon S3 bucket.

> **Note**
>
> You can connect an Amazon Kendra Web Crawler data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

If you store your seed URLs or sitemaps in an Amazon S3 bucket, you must add this permission to the role.

```
,
{"Effect": "Allow",
     "Action": [
         "s3:GetObject"
      ],
      "Resource": [
         "arn:aws:s3:::bucket-name/*"
      ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Box data sources

When you use Box, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Slack.
- Permission to call the required public APIs for the Box connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect a Box data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:PutPrincipalMapping",
                "kendra:DeletePrincipalMapping",
                "kendra:ListGroupsOlderThanOrderingId",
                "kendra:DescribePrincipalMapping"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:123456789012:index/index-id",
                "arn:aws:kendra:us-east-1:123456789012:index/index-id/data-
source/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
```

```
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Confluence data sources

IAM roles for Confluence Connector v1.0

When you use a Confluence server as a data source, you provide a role with the following policies:

- Permission to access the AWS Secrets Manager secret that contains the credentials necessary to connect to Confluence. For more information about the contents of the secret, see Confluence data sources.
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by Secrets Manager.
- Permission to use the `BatchPutDocument` and `BatchDeleteDocument` operations to update the index.

> **Note**
>
> You can connect a Confluence data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
```

```
                "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
            }
        ]
}
```

If you are using a VPC, provide a policy that gives Amazon Kendra access to the required resources. See IAM roles for data sources, VPC for the required policy.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Confluence Connector v2.0

For a Confluence connector v2.0 data source, you provide a role with the following policies.

- Permission to access the AWS Secrets Manager secret that contains the authentication credentials for Confluence. For more information about the contents of the secret, see Confluence data sources.
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by AWS Secrets Manager.
- Permission to use the `BatchPutDocument` and `BatchDeleteDocument` operations to update the index.

You must also attach a trust policy that allows Amazon Kendra to assume the role.

Note

You can connect a Confluence data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

A role policy to allow Amazon Kendra to connect to Confluence.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:your-region:your-account-id:secret:secret-id"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:your-region:your-account-id:key/key-id"
      ],
      "Condition": {
        "StringLike": {
          "kms:ViaService": [
            "secretsmanager.your-region.amazonaws.com"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kendra:PutPrincipalMapping",
        "kendra:DeletePrincipalMapping",
        "kendra:ListGroupsOlderThanOrderingId",
        "kendra:DescribePrincipalMapping"
```

```
      ],
      "Resource": [
        "arn:aws:kendra:your-region:your-account-id:index/index-id",
        "arn:aws:kendra:your-region:your-account-id:index/index-id/data-source/*"
      ]
    }
    {
      "Effect": "Allow",
      "Action": [
        "kendra:BatchPutDocument",
        "kendra:BatchDeleteDocument"
      ],
      "Resource": "arn:aws:kendra:your-region:your-account-id:index/index-id"
    }
  ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Dropbox data sources

When you use Dropbox, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Dropbox.
- Permission to call the required public APIs for the Dropbox connector.

- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect a Dropbox data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "kendra:PutPrincipalMapping",
                "kendra:DeletePrincipalMapping",
                "kendra:ListGroupsOlderThanOrderingId",
                "kendra:DescribePrincipalMapping"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:111122223333:index/index-id",
                "arn:aws:kendra:us-east-1:111122223333:index/index-id/data-
source/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:111122223333:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
```

```
}
```

IAM roles for Drupal data sources

When you use Drupal, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Drupal.
- Permission to call the required public APIs for the Drupal connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect a Drupal data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret_id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
```

```
                "arn:aws:kms:us-east-1:111122223333:key/key_id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.*.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:PutPrincipalMapping",
                "kendra:DeletePrincipalMapping",
                "kendra:ListGroupsOlderThanOrderingId",
                "kendra:DescribePrincipalMapping"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:111122223333:index/index_id",
                "arn:aws:kendra:us-east-1:111122223333:index/index_id/data-
source/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:111122223333:index/index_id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
```

```
    "Statement":[
       {
          "Effect":"Allow",
          "Principal":{
             "Service":"kendra.amazonaws.com"
          },
          "Action":"sts:AssumeRole"
       }
    ]
 }
```

IAM roles for GitHub data sources

When you use GitHub, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your GitHub.
- Permission to call the required public APIs for the GitHub connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect a GitHub data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
```

```
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:PutPrincipalMapping",
                "kendra:DeletePrincipalMapping",
                "kendra:ListGroupsOlderThanOrderingId",
                "kendra:DescribePrincipalMapping"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:123456789012:index/index-id",
                "arn:aws:kendra:us-east-1:123456789012:index/index-id/data-
source/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Gmail data sources

When you use Gmail, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Gmail.
- Permission to call the required public APIs for the Gmailconnector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

Note

You can connect a Gmail data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:PutPrincipalMapping",
                "kendra:DeletePrincipalMapping",
                "kendra:ListGroupsOlderThanOrderingId",
                "kendra:DescribePrincipalMapping"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:123456789012:index/index-id",
                "arn:aws:kendra:us-east-1:123456789012:index/index-id/data-
source/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
```

```
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Google Drive data sources

When you use a Google Workspace Drive data source, you provide Amazon Kendra with a role that has the permissions necessary for connecting to the site. These include:

- Permission to get and decrypt the AWS Secrets Manager secret that contains the client account email, admin account email, and private key necessary to connect to the Google Drive site. For more information about the contents of the secret, see Google Drive data sources.
- Permission to use the BatchPutDocument and BatchDeleteDocument APIs.

Note

You can connect a Google Drive data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

The following IAM policy provides the necessary permissions:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
```

```
                    "kendra:BatchDeleteDocument"
                ],
                "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
            }
        ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for IBM DB2 data sources

When you use an IBM DB2 data source connector, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your IBM DB2 data source instance.
- Permission to call the required public APIs for the IBM DB2 data source connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

Note

You can connect an IBM DB2 data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Jira data sources

When you use Jira, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Jira.
- Permission to call the required public APIs for the Jira connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect a Jira data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:PutPrincipalMapping",
                "kendra:DeletePrincipalMapping",
                "kendra:ListGroupsOlderThanOrderingId",
```

```
                "kendra:DescribePrincipalMapping"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:123456789012:index/index-id",
                "arn:aws:kendra:us-east-1:123456789012:index/index-id/data-
source/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Microsoft Exchange data sources

When you use a Microsoft Exchange data source, you provide Amazon Kendra with a role that has the permissions necessary for connecting to the site. These include:

- Permission to get and decrypt the AWS Secrets Manager secret that contains the application ID and secret key necessary to connect to the Microsoft Exchange site. For more information about the contents of the secret, see Microsoft Exchange data sources.
- Permission to use the BatchPutDocument and BatchDeleteDocument APIs.

> **Note**
>
> You can connect a Microsoft Exchange data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

The following IAM policy provides the necessary permissions:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
```

```
                "Condition": {
                    "StringLike": {
                        "kms:ViaService": [
                            "secretsmanager.us-east-1.amazonaws.com"
                        ]
                    }
                }
            },
            {
                "Effect": "Allow",
                "Action": [
                    "kendra:BatchPutDocument",
                    "kendra:BatchDeleteDocument"
                ],
                "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
            }
        ]
    }
```

If you are storing the list of users to index in an Amazon S3 bucket, you must also provide permission to use the S3 `GetObject` operation. The following IAM policy provides the necessary permissions:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Action": [
                "s3:GetObject"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::bucket-name/*"
            ],
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-ids"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com",
                        "s3.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
```

```
            "Effect":"Allow",
            "Principal":{
               "Service":"kendra.amazonaws.com"
            },
            "Action":"sts:AssumeRole"
         }
      ]
}
```

IAM roles for Microsoft OneDrive data sources

When you use a Microsoft OneDrive data source, you provide Amazon Kendra with a role that has the permissions necessary for connecting to the site. These include:

- Permission to get and decrypt the AWS Secrets Manager secret that contains the application ID and secret key necessary to connect to the OneDrive site. For more information about the contents of the secret, see Microsoft OneDrive data sources.
- Permission to use the BatchPutDocument and BatchDeleteDocument APIs.

> **Note**
>
> You can connect a Microsoft OneDrive data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

The following IAM policy provides the necessary permissions:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
```

```
                ]
            },
            {
                "Effect": "Allow",
                "Action": [
                    "kms:Decrypt"
                ],
                "Resource": [
                    "arn:aws:kms:us-east-1:123456789012:key/key-id"
                ],
                "Condition": {
                    "StringLike": {
                        "kms:ViaService": [
                            "secretsmanager.us-east-1.amazonaws.com"
                        ]
                    }
                }
            },
            {
                "Effect": "Allow",
                "Action": [
                    "kendra:BatchPutDocument",
                    "kendra:BatchDeleteDocument"
                ],
                "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
            }
        ]
}
```

If you are storing the list of users to index in an Amazon S3 bucket, you must also provide permission to use the S3 `GetObject` operation. The following IAM policy provides the necessary permissions:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name/*"
            ],
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-ids"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com",
                        "s3.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Microsoft SharePoint data sources

IAM roles for SharePoint Connector v1.0

For a Microsoft SharePoint connector v1.0 data source, you provide a role with the following policies.

- Permission to access the AWS Secrets Manager secret that contains the user name and password for the SharePoint site. For more information about the contents of the secret, see Microsoft SharePoint data sources.
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by AWS Secrets Manager.
- Permission to use the `BatchPutDocument` and `BatchDeleteDocument` operations to update the index.
- Permission to access the Amazon S3 bucket that contains the SSL certificate used to communicate with the SharePoint site.

You must also attach a trust policy that allows Amazon Kendra to assume the role.

> **Note**
>
> You can connect a Microsoft SharePoint data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:123456789012:index/index-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "kendra.us-east-1.amazonaws.com"
```

```
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name/*"
            ]
        }
    ]
}
```

If you have encrypted the Amazon S3 bucket that contains the SSL certificate used to communicate with the SharePoint site, provide a policy to give Amazon Kendra access to the key.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ]
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for SharePoint Connector v2.0

For a Microsoft SharePoint connector v2.0 data source, you provide a role with the following policies.

- Permission to access the AWS Secrets Manager secret that contains the authentication credentials for the SharePoint site. For more information about the contents of the secret, see Microsoft SharePoint data sources.
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by AWS Secrets Manager.
- Permission to use the `BatchPutDocument` and `BatchDeleteDocument` operations to update the index.
- Permission to access the Amazon S3 bucket that contains the SSL certificate used to communicate with the SharePoint site.

You must also attach a trust policy that allows Amazon Kendra to assume the role.

> **Note**
>
> You can connect a Microsoft SharePoint data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

If you have encrypted the Amazon S3 bucket that contains the SSL certificate used to communicate with the SharePoint site, provide a policy to give Amazon Kendra access to the key.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Microsoft SQL Server data sources

When you use Microsoft SQL Server, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Microsoft SQL Server instance.
- Permission to call the required public APIs for the Microsoft SQL Server connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect a Microsoft SQL Server data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Microsoft Teams data sources

When you use a Microsoft Teams data source, you provide Amazon Kendra with a role that has the permissions necessary for connecting to the site. These include:

- Permission to get and decrypt the AWS Secrets Manager secret that contains the client ID and client secret necessary to connect to Microsoft Teams. For more information about the contents of the secret, see Microsoft Teams data sources.

> **Note**
>
> You can connect a Microsoft Teams data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

The following IAM policy provides the necessary permissions:

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Microsoft Yammer data sources

When you use a Microsoft Yammer data source, you provide Amazon Kendra with a role that has the permissions necessary for connecting to the site. These include:

- Permission to get and decrypt the AWS Secrets Manager secret that contains the application ID and secret key necessary to connect to the Microsoft Yammer site. For more information about the contents of the secret, see Microsoft Yammer data sources.
- Permission to use the BatchPutDocument and BatchDeleteDocument APIs.

> **Note**
>
> You can connect a Microsoft Yammer data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

The following IAM policy provides the necessary permissions:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

If you are storing the list of users to index in an Amazon S3 bucket, you must also provide permission to use the S3 `GetObject` operation. The following IAM policy provides the necessary permissions:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name/*"
            ],
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-ids"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com",
                        "s3.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for MySQL data sources

When you use a My SQL data source connector, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your My SQL data source instance.
- Permission to call the required public APIs for the My SQL data source connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect a MySQL data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Oracle data sources

When you use a Oracle data source connector, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Oracle data source instance.
- Permission to call the required public APIs for the Oracle data source connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

Note

You can connect a Oracle data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for PostgreSQL data sources

When you use a PostgreSQL data source connector, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your PostgreSQL data source instance.
- Permission to call the required public APIs for the PostgreSQL data source connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect a PostgreSQL data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Quip data sources

When you use Quip, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Quip.
- Permission to call the required public APIs for the Quip connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect a Quip data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:PutPrincipalMapping",
                "kendra:DeletePrincipalMapping",
                "kendra:ListGroupsOlderThanOrderingId",
```

```
                "kendra:DescribePrincipalMapping"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:123456789012:index/your-index-id",
                "arn:aws:kendra:us-east-1:123456789012:index/your-index-id/data-
source/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Salesforce data sources

When you use a Salesforce as a data source, you provide a role with the following policies:

- Permission to access the AWS Secrets Manager secret that contains the user name and password for the Salesforce site. For more information about the contents of the secret, see Salesforce data sources.
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by Secrets Manager.
- Permission to use the `BatchPutDocument` and `BatchDeleteDocument` operations to update the index.

> **Note**
>
> You can connect a Salesforce data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
```

```
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for ServiceNow data sources

When you use a ServiceNow as a data source, you provide a role with the following policies:

- Permission to access the Secrets Manager secret that contains the user name and password for the ServiceNow site. For more information about the contents of the secret, see ServiceNow data sources.
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by Secrets Manager.
- Permission to use the `BatchPutDocument` and `BatchDeleteDocument` operations to update the index.

> **Note**
>
> You can connect a ServiceNow data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
```

```
                              "secretsmanager.us-east-1.amazonaws.com"
                          ]
                      }
                  }
              },
              {
                  "Effect": "Allow",
                  "Action": [
                      "kendra:BatchPutDocument",
                      "kendra:BatchDeleteDocument"
                  ],
                  "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
              }
          ]
      }
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Slack data sources

When you use Slack, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Slack.
- Permission to call the required public APIs for the Slack connector.

- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect a Slack data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "kendra:PutPrincipalMapping",
                "kendra:DeletePrincipalMapping",
                "kendra:ListGroupsOlderThanOrderingId",
                "kendra:DescribePrincipalMapping"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:123456789012:index/index-id",
                "arn:aws:kendra:us-east-1:123456789012:index/index-id/data-
source/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
```

```
}
```

IAM roles for Zendesk data sources

When you use Zendesk, you provide a role with the following policies.

- Permission to access your AWS Secrets Manager secret to authenticate your Zendesk Suite.
- Permission to call the required public APIs for the Zendesk connector.
- Permission to call the `BatchPutDocument`, `BatchDeleteDocument`, `PutPrincipalMapping`, `DeletePrincipalMapping`, `DescribePrincipalMapping`, and `ListGroupsOlderThanOrderingId` APIs.

> **Note**
>
> You can connect a Zendesk data source to Amazon Kendra through Amazon VPC. If you are using a Amazon VPC, you need to add additional permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
```

```
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:PutPrincipalMapping",
                "kendra:DeletePrincipalMapping",
                "kendra:ListGroupsOlderThanOrderingId",
                "kendra:DescribePrincipalMapping"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:123456789012:index/index-id",
                "arn:aws:kendra:us-east-1:123456789012:index/index-id/data-
source/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/index-id"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
```

```
    "Statement":[
       {
          "Effect":"Allow",
          "Principal":{
             "Service":"kendra.amazonaws.com"
          },
          "Action":"sts:AssumeRole"
       }
    ]
 }
```

Virtual private cloud (VPC) IAM role

If you use a virtual private cloud (VPC) to connect to your data source, you must provide the following additional permissions.

VPC IAM role

```
{
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:{{region}}:{{account_id}}:subnet/[[subnet_ids]]",
        "arn:aws:ec2:{{region}}:{{account_id}}:security-group/[[security_group]]"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*",
      "Condition": {
        "StringLike": {
          "aws:RequestTag/AWS_KENDRA": "kendra_{{account_id}}_{{index_id}}_*"
        }
      }
    },
```

```
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkInterface"
        }
      }
    },

{
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterfacePermission"
      ],
      "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/AWS_KENDRA": "kendra_{{account_id}}_{{index_id}}_*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:DescribeVpcs",
        "ec2:DescribeRegions",
        "ec2:DescribeNetworkInterfacePermissions",
        "ec2:DescribeSubnets"
      ],
      "Resource": "*"
    }
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for frequently asked questions (FAQs)

When you use the CreateFaq API to load questions and answers into an index, you must provide Amazon Kendra with an IAM role with access to the Amazon S3 bucket that contains the source files. If the source files are encrypted, you must provide permission to use the AWS KMS customer master key (CMK) to decrypt the files.

IAM roles for FAQs

A required role policy to allow Amazon Kendra to access an Amazon S3 bucket.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name/*"
            ]
        }
```

```
    ]
}
```

An optional role policy to allow Amazon Kendra to use an AWS KMS customer master key (CMK) to decrypt files in an Amazon S3 bucket.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "kendra.us-east-1.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
```

```
            "Principal":{
               "Service":"kendra.amazonaws.com"
            },
            "Action":"sts:AssumeRole"
         }
      ]
}
```

IAM roles for query suggestions

When you use an Amazon S3 file as a query suggestions block list, you supply a role that has permission to access the Amazon S3 file and the Amazon S3 bucket. If the block list text file (the Amazon S3 file) in the Amazon S3 bucket is encrypted, you must provide permission to use the AWS KMS customer master key (CMK) to decrypt the documents.

IAM roles for query suggestions

A required role policy to allow Amazon Kendra to use the Amazon S3 file as your query suggestions block list.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {"Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name/*"
            ]
        }
    ]
}
```

An optional role policy to allow Amazon Kendra to use an AWS KMS customer master key (CMK) to decrypt documents in an Amazon S3 bucket.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ]
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for principal mapping of users and groups

When you use the PutPrincipalMapping API to map users to their groups for filtering search results by user context, you need to provide a list of users or sub groups that belong to a group. If your list is more than 1000 users or sub groups for a group, you need to supply a role that has permission to access the Amazon S3 file of your list and the Amazon S3 bucket. If the text file (the Amazon S3 file) of the list in the Amazon S3 bucket is encrypted, you must provide permission to use the AWS KMS customer master key (CMK) to decrypt the documents.

IAM roles for principal mapping

A required role policy to allow Amazon Kendra to use the Amazon S3 file as your list of users and sub groups that belong to a group.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {"Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name/*"
            ]
        }
    ]
}
```

An optional role policy to allow Amazon Kendra to use an AWS KMS customer master key (CMK) to decrypt documents in an Amazon S3 bucket.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
```

```
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ]
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

It is recommended that you include `aws:sourceAccount` and `aws:sourceArn` in the trust policy. This limits permissions and securely checks if `aws:sourceAccount` and `aws:sourceArn` are the same as provided in the IAM role policy for the `sts:AssumeRole` action. This prevents unauthorized entities from accessing your IAM roles and their permissions. For more information, see the AWS Identity and Access Management guide on the confused deputy problem.

IAM roles for AWS IAM Identity Center

When you use the UserGroupResolutionConfiguration object to fetch access levels of groups and users from an AWS IAM Identity Center identity source, you need to supply a role that has permission to access IAM Identity Center.

IAM roles for AWS IAM Identity Center

A required role policy to allow Amazon Kendra to access IAM Identity Center.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sso-directory:SearchUsers",
                "sso-directory:ListGroupsForUser",
                "sso-directory:DescribeGroups",
                "sso:ListDirectoryAssociations"
            ],
            "Resource": [
                 "*"
            ]
        },
        {
          "Sid": "iamPassRole",
          "Effect": "Allow",
          "Action": "iam:PassRole",
          "Resource": "*",
          "Condition": {
            "StringEquals": {
              "iam:PassedToService": [
                "kendra.amazonaws.com"
              ]
            }
          }
        }
     ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

IAM roles for Amazon Kendra experiences

When you use the CreateExperience or UpdateExperience APIs to create or update a search application, you must supply a role that has permission to access the necessary operations and IAM Identity Center.

IAM roles for Amazon Kendra search experience

A required role policy to allow Amazon Kendra to access `Query` operations, `QuerySuggestions` operations, `SubmitFeedback` operations, and IAM Identity Center that stores your user and group information.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsKendraSearchAppToCallKendraApi",
            "Effect": "Allow",
            "Action": [
                "kendra:GetQuerySuggestions",
                "kendra:Query",
                "kendra:DescribeIndex",
```

```
                "kendra:ListFaqs",
                "kendra:DescribeDataSource",
                "kendra:ListDataSources",
                "kendra:DescribeFaq",
                "kendra:SubmitFeedback"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:123456789012:index/index-id"
            ]
        },
        {
            "Sid": "AllowKendraSearchAppToDescribeDataSourcesAndFaq",
            "Effect": "Allow",
            "Action": [
                "kendra:DescribeDataSource",
                "kendra:DescribeFaq"
            ],
            "Resource": [
                "arn:aws:kendra:us-east-1:123456789012:index/index-id/data-
source/data-source-id",
                "arn:aws:kendra:us-east-1:123456789012:index/index-id/faq/faq-id"
            ]
        },
        {
            "Sid": "AllowKendraSearchAppToCallSSODescribeUsersAndGroups",
            "Effect": "Allow",
            "Action": [
                "sso-directory:ListGroupsForUser",
                "sso-directory:SearchGroups",
                "sso-directory:SearchUsers",
                "sso-directory:DescribeUser",
                "sso-directory:DescribeGroup",
                "sso-directory:DescribeGroups",
                "sso-directory:DescribeUsers",
                "sso:ListDirectoryAssociations"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "kendra.us-east-1.amazonaws.com"
                    ]
```

```
                }
            }
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

It is recommended that you include `aws:sourceAccount` and `aws:sourceArn` in the trust policy. This limits permissions and securely checks if `aws:sourceAccount` and `aws:sourceArn` are the same as provided in the IAM role policy for the `sts:AssumeRole` action. This prevents unauthorized entities from accessing your IAM roles and their permissions. For more information, see the AWS Identity and Access Management guide on the confused deputy problem.

IAM roles for Custom Document Enrichment

When you use the CustomDocumentEnrichmentConfiguration object to apply advanced alterations of your document metadata and content, you must supply a role that has the required permissions to run `PreExtractionHookConfiguration` and/or `PostExtractionHookConfiguration`. You configure a Lambda function for `PreExtractionHookConfiguration` and/or `PostExtractionHookConfiguration` to apply advanced alterations of your document metadata and content during the ingestion process. If you choose to activate Server Side

Encryption for your Amazon S3 bucket, you must provide permission to use the AWS KMS customer master key (CMK) to encrypt and decrypt the objects stored in your Amazon S3 bucket.

IAM roles for Custom Document Enrichment

A required role policy to allow Amazon Kendra to run `PreExtractionHookConfiguration` and `PostExtractionHookConfiguration` with encryption for your Amazon S3 bucket.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name"
            ],
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:123456789012:key/key-id"
            ]
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": "arn:aws:lambda:us-east-1:123456789012:function:lambda-
function"
        }
    ]
}
```

An optional role policy to allow Amazon Kendra to run `PreExtractionHookConfiguration` and `PostExtractionHookConfiguration` without encryption for your Amazon S3 bucket.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name"
            ],
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "lambda:InvokeFunction"
            ],
            "Resource": "arn:aws:lambda:us-east-1:123456789012:function:lambda-function"
        }
    ]
}
```

A trust policy to allow Amazon Kendra to assume a role.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"kendra.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

It is recommended that you include `aws:sourceAccount` and `aws:sourceArn` in the trust policy. This limits permissions and securely checks if `aws:sourceAccount` and `aws:sourceArn` are the same as provided in the IAM role policy for the `sts:AssumeRole` action. This prevents unauthorized entities from accessing your IAM roles and their permissions. For more information, see the AWS Identity and Access Management guide on the confused deputy problem.

Deploying Amazon Kendra

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

When it comes time to deploy Amazon Kendra search to your website, we provide source code that you can use with React to get a head start on your application. The source code is provided with no charge under a modified MIT license. You can use it as is or change it for your own needs. The provided React app is an example to help you get started. It's not a production ready app.

To deploy a search application with no code and generate an endpoint URL to your search page with access control, see Amazon Kendra Experience Builder.

The following example code adds Amazon Kendra search to an existing React web application:

- https://kendrasamples.s3.amazonaws.com/kendrasamples-react-app.zip—Sample files that developers can use to build a functional search experience into their existing React web application.

The examples are modeled after the search page of the Amazon Kendra console. They have the same features for searching and displaying search results. You can use the whole example, or you can choose just one of the features for your own use.

To see the three components of the search page in the Amazon Kendra console, choose the code icon (**</>**) from the right menu. Hover your pointer over each section to see a brief description of the component and to get the URL of the component's source.

Topics

- Results component
- Facets component
- Pagination component
- Building a search experience with no code

Overview

You add the example code to an existing React web application to activate search. The example code includes a Readme file with steps to set up a new React development environment. The example data in the code example can be used to demonstrate a search. The search files and components in the example code are structured as follows:

- Main search page (`Search.tsx`)—This is the main page that contains all of the components. This is where you integrate your application with the Amazon Kendra API.
- Search bar—This is the component where a user enters a search term and calls the search function.
- Results—This is the component that displays the results from Amazon Kendra. It has three components: Suggested answers, FAQ results, and recommended documents.
- Facets—This is the component that shows the facets in the search results and allows you to choose a facet to narrow the search.
- Pagination—This is the component that paginates the response from Amazon Kendra.

Prerequisites

Before you begin, you need the following:

- Node.js and npm installed. Node.js version 19 or older is required.
- Python 3 or Python 2 downloaded and installed.
- SDK for Java or AWS SDK for JavaScript to make API calls to Amazon Kendra.
- An existing React web application. The example code includes a Readme file with steps on how to set up a new React development environment, including using required frameworks/libraries. You can also follow the quick start instructions in the React documentation on creating a React web app.

- The required libraries and dependencies configured in your development environment. The example code includes a Readme file that lists the required libraries and package dependencies. Note that `sass` is required, as `node-sass` is deprecated. If you previously installed `node-sass`, uninstall this and install `sass`.

Setting up the example

A complete procedure for adding Amazon Kendra search to a React application is in the Readme file included in the code example.

To get started using kendrasamples-react-app.zip

1. Make sure you have completed the Prerequisites, including downloading and installing Node.js and npm.
2. Download kendrasamples-react-app.zip and unzip.
3. Open your terminal and go to `aws-kendra-example-react-app/src/services/`. Open `local-dev-credentials.json` and provide your credentials. Do not add this file to any public repository.
4. Go to `aws-kendra-example-react-app` and install the dependencies in `package.json`. Run `npm install`.
5. Launch a demo version of your app on your local server. Run `npm start`. You can stop the local server by entering on your keyboard `Cmd/Ctrl + C`.
6. You can change the port or host (for example, IP address) by going to `package.json` and update the host and port: `"start": "HOST=[host] PORT=[port] react-scripts start"`. If you use Windows: `"start": "set HOST=[host] && set PORT=[port] && react-scripts start"`.
7. If you have a registered website domain, you can specify this in `package.json` after your app name. For example, `"homepage": "https://mywebsite.com"`. You must run `npm install` again to update new dependencies, and then run `npm start`.
8. To build the app, run `npm build`. Upload the contents of the build directory to your hosting provider.

> **Warning**
>
> The React app is **not** production ready. It's an example of deploying an app for Amazon Kendra search.

Main search page

The main search page (`Search.tsx`) contains all of the example search components. It includes the search bar component for output, the results components to display the response from the [Query](#) API, and a pagination component for paging through the response.

Search component

The search component provides a text box to enter query text. The `onSearch` function is a hook that calls the main function in `Search.tsx` to make the Amazon Kendra [Query](#) API call.

Results component

The results component shows the response from the `Query` API. The results are shown in three separate areas.

- Suggested answers—These are the top results returned by the `Query` API. It contains up to three suggested answers. In the response, they have the result type `ANSWER`.
- FAQ answers—These are the frequently asked questions results returned by the response. FAQs are added to the index separately. In the response, they have the type `QUESTION_ANSWER`. For more information, see [Questions and answers](#).
- Recommended documents—These are additional documents that Amazon Kendra returns in the response. In the response from the `Query` API, they have the type `DOCUMENT`.

The results components share a set of components for features like highlighting, titles, links, and more. The shared components must be present for the result components to work.

Facets component

The facets component lists the facets available in the search results. Each facet classifies the response along a specific dimension, such as author. You can refine the search to a specific facet by choosing one from the list.

After you select a facet, the component calls `Query` with an attribute filter that restricts the search to documents that match the facet.

Pagination component

The pagination component allows you to display the search results from the `Query` API in multiple pages. It calls the `Query` API with the `PageSize` and `PageNumber` parameters to get a specific page of results.

Building a search experience with no code

You can build and deploy an Amazon Kendra search application without the need for any front-end code. Amazon Kendra *Experience Builder* helps you build and deploy a fully functional search application in a few clicks so that you can start searching right away. You can custom design your search page and tune your search to tailor the experience to your users' needs. Amazon Kendra generates a unique, fully hosted endpoint URL of your search page to start searching your documents and FAQs. You can quickly build a proof of concept of your search experience and share it with others.

You use the search experience template available in the builder to customize your search. You can invite others to collaborate in building your search experience, or evaluate search results for tuning purposes. Once your search experience is ready for your users to start searching, you simply share the secure endpoint URL.

How the search Experience Builder works

The overall process of building a search experience is as follows:

1. You create your search experience by giving it a name, description, and choosing your data sources you want to use for your search experience.

2. You configure your list of users and groups in AWS IAM Identity Center and then assign them access rights to your search experience. You include yourself as an owner of the experience. For more information, see the section called "Providing access to your search page".
3. You open the Amazon Kendra Experience Builder to design and tune your search page. You can share your endpoint URL of your search experience with others who you assign own-edit access rights or view-search access rights.

You call the CreateExperience API to create and configure your search experience. If you use the console, you select your index and then select **Experiences** in the navigation menu to configure your experience.

Design and tune your search experience

Once you create and configure your search experience, you open the search experience using an endpoint URL to start customizing your search as an owner with editor access rights. You type your query into the search box, then customize your search using the editing options on the side panel to see how they apply to your page. When you are ready to publish, select **Publish**. You can also toggle between **Switch to live view**, to view the latest published version of your search page, and **Switch to build mode**, to edit or customize your search page.

The following are ways you can customize your search experience.

Filter

Add faceted search or filter by document attributes. This includes custom attributes. You can add a filter using your own configured metadata fields. For example, to facet search by each city category, use a `_category` custom document attribute that contains all the city categories.

Suggested answer

Add machine learning generated answers to your users' queries. For example, *'How difficult is this course?'*. Amazon Kendra can retrieve the most relevant text across all documents referring to a course's difficulty and suggest the most relevant answer.

FAQ

Add a FAQ document to provide answers to frequently asked questions. For example, *'How many hours to complete this course?'*. Amazon Kendra can use the FAQ document containing the answer to this question and give the correct answer.

Sort

Add sorting of the search results so that your users can organize the results by relevancy, created time, last updated time, and other sorting criteria.

Documents

Configure how documents or search results are displayed on your search page. You can configure how many results display on the page, include pagination such as page numbers, activate a user feedback button, and arrange how document metadata fields are displayed in a search result.

Language

Select a language to filter the search results or documents in the selected language.

Search box

Configure the size and placeholder text of your search box, as well as allow query suggestions.

Relevance tuning

Add boosting to document metadata fields to place more weight on these fields when your users search for documents. You can add a weight that starts at 1 and incrementally increases to 10. You can boost text, date, and numeric field types. For example, to give `_last_updated_at` and `_created_at` more weight or importance than other fields, give these fields a weight of 1 to 10, depending on their importance. You can apply different relevance tuning configurations for each search application or experience.

Providing access to your search page

Access to your search experience is through IAM Identity Center. When you configure your search experience, you grant other people listed in your Identity Center directory access to your Amazon Kendra search page. They receive an email that directs them to sign in using their credentials in IAM Identity Center to access the search page. You must set up IAM Identity Center at the organization level or account holder level in AWS Organizations. For more information on setting up IAM Identity Center, see [Getting started with IAM Identity Center](#).

You activate user identities in IAM Identity Center with your search experience and assign *Viewer* or *Owner* access permissions using the API or the console.

- **Viewer**: Allowed to issue queries, receive suggested answers relevant to their search, and contribute their feedback to Amazon Kendra so that it keeps improving the search.
- **Owner**: Allowed to customize the design of the search page, tune the search, and use the search application as a *Viewer*. Disabling access to viewers in the console is currently not supported.

To assign other people access to your search experience, you first activate user identities in IAM Identity Center with your Amazon Kendra experience by using the ExperienceConfiguration object. You specify the field name that contains the identifiers of your users such as user name or email address. You then grant your list of users access to your search experience using the AssociateEntitiesToExperience API and define their permissions as *Viewer* or *Owner* using the AssociatePersonasToEntities API. You specify each user or group using the EntityConfiguration object and whether that user or group is a *Viewer* or *Owner* using the EntityPersonaConfiguraton object.

To assign other people access to your search experience using the console, you first need to create an experience and confirm your identity and that you are an owner. Then you can assign other users or groups as viewers or owners. In the console, select your index and then select **Experiences** in the navigation menu. After you create your experience, you can select your experience from the list. Go to **Access management** to assign users or groups as viewers or owners.

Configuring a search experience

The following is an example of configuring or creating a search experience.

Console

To create an Amazon Kendra search experience

1. In the left navigation pane, under **Indexes**, select **Experiences** and then select **Create experience**.
2. On the **Configure experience** page, enter a name and description for your experience, choose your content sources, and choose the IAM role for your experience. For more information on IAM roles, see IAM roles for Amazon Kendra experiences.
3. On the **Confirm your identity from an Identity Center directory** page, select your user ID such as your email. If you do not have an Identity Center directory, simply enter your full name and email to create an Identity Center directory. This includes you as a user of the experience and automatically assigns you owner access rights.

4. On the **Review to open Experience Builder** page, review your configuration details and select **Create experience and open Experience Builder** to start editing your search page.

CLI

To create an Amazon Kendra experience

```
aws kendra create-experience \
 --name experience-name \
 --description "experience description" \
 --index-id index-id \
 --role-arn arn:aws:iam::account-id:role/role-name \
 --configuration '{"ExperienceConfiguration":[{"ContentSourceConfiguration":
{"DataSourceIds":["data-source-1","data-source-2"]},
 "UserIdentityConfiguration":"identity attribute name"}]}'

aws kendra describe-experience \
 --endpoints experience-endpoint-URL(s)
```

Python

To create an Amazon Kendra experience

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Create an experience.")

# Provide a name for the experience
name = "experience-name"
# Provide an optional description for the experience
description = "experience description"
# Provide the index ID for the experience
index_id = "index-id"
# Provide the IAM role ARN required for Amazon Kendra experiences
role_arn = "arn:aws:iam::${account-id}:role/${role-name}"
# Configure the experience
configuration = {"ExperienceConfiguration":
        [{
```

```
            "ContentSourceConfiguration":{"DataSourceIds":["data-source-1","data-
source-2"]},
            "UserIdentityConfiguration":"identity attribute name"
        }]
    }

try:
    experience_response = kendra.create_experience(
        Name = name,
        Description = description,
        IndexId = index_id,
        RoleArn = role_arn,
        Configuration = configuration
    )

    pprint.pprint(experience_response)

    experience_endpoints = experience_response["Endpoints"]

    print("Wait for Amazon Kendra to create the experience.")

    while True:
        # Get the details of the experience, such as the status
        experience_description = kendra.describe_experience(
            Endpoints = experience_endpoints
        )
        status = experience_description["Status"]
        print(" Creating experience. Status: "+status)
        time.sleep(60)
        if status != "CREATING":
            break

except  ClientError as e:
        print("%s" % e)

print("Program ends.")
```

Java

To create an Amazon Kendra

```
package com.amazonaws.kendra;

import java.util.concurrent.TimeUnit;
```

```
import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.CreateExperienceRequest;
import software.amazon.awssdk.services.kendra.model.CreateExperienceResponse;
import software.amazon.awssdk.services.kendra.model.DescribeExperienceRequest;
import software.amazon.awssdk.services.kendra.model.DescribeExperienceResponse;
import software.amazon.awssdk.services.kendra.model.ExperienceStatus;


public class CreateExperienceExample {

    public static void main(String[] args) throws InterruptedException {
        System.out.println("Create an experience");

        String experienceName = "experience-name";
        String experienceDescription = "experience description";
        String indexId = "index-id";
        String experienceRoleArn = "arn:aws:iam::account-id:role/role-name";

        KendraClient kendra = KendraClient.builder().build();

        CreateExperienceRequest createExperienceRequest = CreateExperienceRequest
            .builder()
            .name(experienceName)
            .description(experienceDescription)
            .roleArn(experienceRoleArn)
            .configuration(
                ExperienceConfiguration
                    .builder()
                    .contentSourceConfiguration(
                        ContentSourceConfiguration(
                            .builder()
                            .dataSourceIds("data-source-1","data-source-2")
                            .build()
                        )
                    )
                    .userIdentityConfiguration(
                        UserIdentityConfiguration(
                            .builder()
                            .identityAttributeName("identity-attribute-name")
                            .build()
                        )
                    ).build()
            ).build();
```

```
        CreateExperienceResponse createExperienceResponse =
 kendra.createExperience(createExperienceRequest);
        System.out.println(String.format("Experience response %s",
 createExperienceResponse));

        String experienceEndpoints = createExperienceResponse.endpoints();

        System.out.println(String.format("Wait for Kendra to create the
 experience.", experienceEndpoints));
        while (true) {
            DescribeExperienceRequest describeExperienceRequest =
 DescribeExperienceRequest.builder().endpoints(experienceEndpoints).build();
            DescribeExperienceResponse describeEpxerienceResponse =
 kendra.describeExperience(describeExperienceRequest);
            ExperienceStatus status = describeExperienceResponse.status();
            TimeUnit.SECONDS.sleep(60);
            if (status != ExperienceStatus.CREATING) {
                break;
            }
        }

        System.out.println("Experience creation is complete.");
    }
}
```

Adjusting capacity

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

Amazon Kendra provides resources for your index in *capacity units*. Each capacity unit provides additional resources for your index. There are separate capacity units for document storage and for queries. You can only add capacity units to GenAI Enterprise Edition and Amazon Kendra Enterprise indices. You can't add capacity to a Developer Edition index.

A document storage capacity unit provides the following additional storage for your index.

- **Amazon Kendra GenAI Enterprise Edition** – 20,000 documents or 200MB of extracted text.
- **Amazon Kendra Enterprise Edition** – 100,000 documents or 30 GB of storage.

A query capacity unit provides the following additional queries for your index. Query per second is shared across retrieve and query API.

- **Amazon Kendra GenAI Enterprise Edition** – 0.1 queries per second or approximately 8,000 queries per day.
- **Amazon Kendra Enterprise Edition** – 0.1 queries per second or approximately 8,000 queries per day.

Each index comes with a base capacity equal to 1 capacity unit (30 GB/200MB of storage and 0.1 queries per second). There is an additional cost for each additional capacity unit. For details, see Amazon Kendra pricing.

You can add up to 100 extra capacity units to your storage and query resources for an index. If you need more units, simply contact Support.

You can adjust capacity units up to 5 times per day to fit your usage requirements. You can't reduce document storage capacity below the number of documents stored in your index. For example, if you are storing 150,000 documents, you can't reduce the storage capacity below 1 additional unit.

You can view the resources an index is using in the console by selecting the name of the index to open the index settings and other information, or you can use the DescribeIndex API.

Amazon Kendra also returns exceptions when you exceed the capacity of an index. You get a `ServiceQuotaExceededException` when the total extracted size of all the documents exceeds the limit for an index. You get a `InvalidRequest` for each document when the number of documents exceeds the limit for an index. You get a `ThrottlingException` when the number of queries per second exceeds the limit. For more information on limits, see Quotas for Amazon Kendra.

Accumulated queries will last up to 24 hours.

Viewing capacity

View the resources that your index is using with the Amazon Kendra console by selecting the name of your index to access the details. The console also provides usage graphs so you can determine how much storage and query capacity your index uses. You can use this information to help you plan when to add additional capacity.

To view document storage and query use (console)

1. Sign into the AWS Management Console and open the Amazon Kendra console at https://console.aws.amazon.com/kendra/home.
2. From the list of indexes, choose the index you want to access.
3. Scroll to the settings section to view the current total document storage and query capacity.

To view capacity using the Amazon Kendra API, use the `CapacityUnits` parameter in the DescribeIndex API.

Adding and removing capacity

If you need additional capacity for your index, you can add it using the console or the Amazon Kendra API.

To add or remove storage or query capacity (console)

1. Sign into the AWS Management Console and open the Amazon Kendra console at https://console.aws.amazon.com/kendra/home.

2. From the list of indexes, choose the index that you want to access.
3. Select **Edit**, or select **Edit** from the **Actions** dropdown.
4. Select **Next** to go to the provisioning details page.
5. Add or remove document storage and/or query capacity units.
6. Continue to select **Next** to go to the review page and then select **Update** to save your changes.

After you update the capacity of your index, it can take several minutes for the changes to take effect.

To add or remove capacity using the Amazon Kendra API, use the `CapacityUnits` parameter in the UpdateIndex API.

Amazon Kendra Intelligent Ranking capacity

A capacity unit provides the following additional rescore requests per second for a rescore execution plan. A rescore execution plan is a resource used to provision the Rescore API.

- 0.01 requests per second.

Each rescore execution plan comes with a base capacity equal to 1 capacity unit (0.01 requests per second). There is an additional cost for each additional capacity unit. For details, see Amazon Kendra pricing.

You can add up to 1000 extra capacity units for a rescore execution plan. If you need more units, simply contact Support.

Query suggestions capacity

When using query suggestions, there's a base query capacity of 2.5 GetQuerySuggestions calls per second. The `GetQuerySuggestions` capacity is five times the provisioned query capacity for an index, or the base capacity of 2.5 calls per second, whichever is higher. For example, the base capacity for an index is 0.1 queries per second, and `GetQuerySuggestions` capacity has a base of 2.5 calls per second. If you add another 0.1 queries per second to total 0.2 queries per second for an index, the `GetQuerySuggestions` capacity is 2.5 calls per second (higher than five times 0.2 queries per second).

Amazon Kendra experience capacity

Search experience capacity

Amazon Kendra starts to throttle `Query`, `QuerySuggestions`, `SubmitFeedback` for your Amazon Kendra experience at 15 requests per second and 40 requests per second for query bursting. For an index with more than 150 query capacity units, these limits still apply.

For example, your query capacity units for your index is 150, so your search experience application can handle 15 requests per second. However, if you scaled to 200 query capacity units, then your search experience app would still only handle 15 requests per second. If you limit your index to 100 query capacity units, then your search experience app would only handle 10 requests per second.

Adaptive query bursting

Amazon Kendra has a provisioned base capacity of 1 query capacity unit. You can use up to 8,000 queries per day with a minimum throughput of 0.1 queries per second (per query capacity unit). Accumulated queries will last up to 24 hours and can accommodate bursts of traffic. The amount of burst allowed varies because it depends on the cluster's load at any given time. Provision enough query capacity units to handle your peak load levels.

An adaptive approach to handling unexpected bursts of traffic beyond the provisioned throughput is Amazon Kendra's built-in *adaptive query bursting*. Adaptive query bursting is available in the Enterprise Edition of Amazon Kendra.

Adaptive query bursting is a built-in capability that allows you to apply unused query capacity to handle unexpected traffic. Amazon Kendra accumulates your unused queries at your provisioned queries per second rate, every second, up to the maximum number of queries you've provisioned for your Amazon Kendra index. These accumulated queries are used for unexpected traffic above the allocated capacity. Optimal performance of adaptive query bursting can vary, depending on several factors such as your total index size, query complexity, accumulated unused queries, and overall load on your index. It is recommended that you perform your own load tests to accurately measure bursting capacity.

Getting started

This section shows you how to create a data source and add your documents to an Amazon Kendra index. Instructions are provided for the AWS console, the AWS CLI, a Python program using the AWS SDK for Python (Boto3), and a Java program using the AWS SDK for Java.

Topics

Prerequisites

The following steps are prequisites for the getting started exercises. The steps show you how to set up your account, create an IAM role that gives Amazon Kendra permission to make calls on your behalf, and index documents from an Amazon S3 bucket. An S3 bucket is used as an example, but you can use other data sources that Amazon Kendra supports. See Data sources.

Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call or text message and entering a verification code on the phone keypad.

When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to a user, and use only the root user to perform tasks that require root user access.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**.

Create a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the AWS Management Console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.
2. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the *IAM User Guide*.

Create a user with administrative access

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.
2. In IAM Identity Center, grant administrative access to a user.

   For a tutorial about using the IAM Identity Center directory as your identity source, see Configure user access with the default IAM Identity Center directory in the *AWS IAM Identity Center User Guide*.

Sign in as the user with administrative access

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

  For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the *AWS Sign-In User Guide*.

Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

   For instructions, see Create a permission set in the *AWS IAM Identity Center User Guide*.
2. Assign users to a group, and then assign single sign-on access to the group.

   For instructions, see Add groups in the *AWS IAM Identity Center User Guide*.

- If you are using an S3 bucket containing documents to test Amazon Kendra, create an S3 bucket in the same region that you are using Amazon Kendra. For instructions, see Creating and Configuring an S3 Bucket in the *Amazon Simple Storage Service User Guide*.

  Upload your documents to your S3 bucket. For instructions, see Uploading, Downloading, and Managing Objects in the *Amazon Simple Storage Service User Guide*.

  If you are using another data source, you must have an active site and credentials to connect to the data source.

If you are using the console to get started, start with Getting started with the Amazon Kendra console.

Amazon Kendra resources: AWS CLI, SDK, console

There are certain permissions required if you use CLI, SDK, or the console.

To use Amazon Kendra for the CLI, SDK, or console you must have permissions to allow Amazon Kendra to create and manage resources on your behalf. Depending on your use case, these permissions include access to the Amazon Kendra API itself, AWS KMS keys if you want to encrypt your data through a custom CMK, Identity Center directory if you want to integrate with AWS IAM

Identity Center or [create a Search Experience](). For a full list of permissions for different use cases, see [IAM roles]().

First, you must attach the below permissions to your IAM user.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1644430853544",
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid": "Stmt1644430878150",
      "Action": "kendra:*",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid": "Stmt1644430973706",
      "Action": [
        "sso:AssociateProfile",
        "sso:CreateManagedApplicationInstance",
        "sso:DeleteManagedApplicationInstance",
        "sso:DisassociateProfile",
        "sso:GetManagedApplicationInstance",
        "sso:GetProfile",
        "sso:ListDirectoryAssociations",
        "sso:ListProfileAssociations",
        "sso:ListProfiles"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid": "Stmt1644430999558",
```

```
      "Action": [
        "sso-directory:DescribeGroup",
        "sso-directory:DescribeGroups",
        "sso-directory:DescribeUser",
        "sso-directory:DescribeUsers"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid": "Stmt1644431025960",
      "Action": [
        "identitystore:DescribeGroup",
        "identitystore:DescribeUser",
        "identitystore:ListGroups",
        "identitystore:ListUsers"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Second, if you use the CLI or SDK, you must also create an IAM role and policy to access Amazon CloudWatch Logs. If you are using the console, you don't need to create an IAM role and policy for this. You create this as part of the console procedure.

To create an IAM role and policy for the AWS CLI and SDK that allows Amazon Kendra to access your Amazon CloudWatch Logs.

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. From the left menu, choose **Policies** and then choose **Create policy**.
3. Choose **JSON** and then replace the default policy with the following:

   JSON

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
   ```

```
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "AWS/Kendra"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:123456789012:log-group:/aws/kendra/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogStreams",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:123456789012:log-group:/aws/kendra/
*:log-stream:*"
            ]
        }
    ]
}
```

4. Choose **Review policy**.

5. Name the policy "KendraPolicyForGettingStartedIndex" and then choose **Create policy**.
6. From the left menu, choose **Roles** and then choose **Create role**.
7. Choose **Another AWS account** and then type your account ID in **Account ID**. Choose **Next: Permissions**.
8. Choose the policy that you created above and then choose **Next: Tags**
9. Don't add any tags. Choose **Next: Review**.
10. Name the role "KendraRoleForGettingStartedIndex" and then choose **Create role**.
11. Find the role that you just created. Choose the role name to open the summary. Choose **Trust relationships** and then choose **Edit trust relationship**.
12. Replace the existing trust relationship with the following:

    JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "kendra.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
}
```

13. Choose **Update trust policy**.

Third, if you use an Amazon S3 to store your documents or you are using S3 to test Amazon Kendra, you also must create an IAM role and policy to access your bucket. If you are using another data source, see IAM roles for data sources.

To create an IAM role and policy that allows Amazon Kendra to access and index your Amazon S3 bucket.

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.

2. From the left menu, choose **Policies** and then choose **Create policy**.
3. Choose **JSON** and then replace the default policy with the following:

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket name/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket name"
            ],
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:us-east-1:123456789012:index/*"
        }
    ]
}
```

4. Choose **Review policy**.
5. Name the policy "KendraPolicyForGettingStartedDataSource" and then choose **Create policy**.
6. From the left menu, choose **Roles** and then choose **Create role**.

7. Choose **Another AWS account** and then type your account ID in **Account ID**. Choose **Next: Permissions**.
8. Choose the policy that you created above and then choose **Next: Tags**
9. Don't add any tags. Choose **Next: Review**.
10. Name the role "KendraRoleForGettingStartedDataSource" and then choose **Create role**.
11. Find the role that you just created. Choose the role name to open the summary. Choose **Trust relationships** and then choose **Edit trust relationship**.
12. Replace the existing trust relationship with the following:

    JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "kendra.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
}
```

13. Choose **Update trust policy**.

Depending on how you want to use the Amazon Kendra API, do one of the following.

- Getting started (AWS CLI)
- Getting started (AWS SDK for Java)
- Getting started (AWS SDK for Python (Boto3))

Getting started with the Amazon Kendra console

The following procedures show how to create and test an Amazon Kendra index by using the AWS console. In the procedures you create an index and a data source for an index. Finally, you test your index by making a search request.

Step 1: To create an index (console)

1. Sign in to the AWS Management Console and open the Amazon Kendra console at https://console.aws.amazon.com/kendra/.
2. Select **Create index** in the **Indexes** section.
3. In the **Specify index details** page, give your index a name and a description.
4. In **IAM role**, choose **Create a new role** and then give the role a name. The IAM role will have the prefix "AmazonKendra-".
5. Leave all of the other fields at their defaults. Choose **Next**.
6. In the **Configure user access control** page, choose **Next**.
7. In the **Provisioning details** page, choose **Developer edition**.
8. Choose **Create** to create your index.
9. Wait for your index to be created. Amazon Kendra provisions the hardware for your index. This operation can take some time.

Step 2: To add a data source to an index (console)

1. View the available data sources to connect Amazon Kendra to and index your documents.
2. In the navigation pane, select **Data sources** and then select **Add data source** for your chosen data source.
3. Follow the steps to configure the data source.

Step 3: To search an index (console)

1. In the navigation pane, choose the option to search your index.
2. Enter a search term that's appropriate for your index. The **top results** and **top document** results are shown.

Getting started (AWS CLI)

The following procedure shows how to create an Amazon Kendra index using the AWS CLI. The procedure creates a data source, index, and runs a query on the index.

To create an Amazon Kendra index (CLI)

1. Do the Prerequisites.
2. Enter the following command to create an index.

```
aws kendra create-index \
 --name cli-getting-started-index \
 --description "Index for CLI getting started guide." \
 --role-arn arn:aws:iam::account id:role/KendraRoleForGettingStartedIndex
```

3. Wait for Amazon Kendra to create the index. Check the progress using the following command. When the status field is `ACTIVE`, go on to the next step.

```
aws kendra describe-index \
 --id index id
```

4. At the command prompt, enter the following command to create a data source.

```
aws kendra create-data-source \
 --index-id index id \
 --name data source name \
 --role-arn arn:aws:iam::account id:role/KendraRoleForGettingStartedDataSource \
 --type S3 \
 --configuration '{"S3Configuration":{"BucketName":"S3 bucket name"}}'
```

If you connect to your data source using a template schema, configure the template schema.

```
aws kendra create-data-source \
 --index-id index id \
 --name data source name \
 --role-arn arn:aws:iam::account id:role/KendraRoleForGettingStartedDataSource \
 --type TEMPLATE \
 --configuration '{"TemplateConfiguration":{"Template":{JSON schema}}}'
```

5. It will take Amazon Kendra a while to create the data source. Enter the following command to check the progress. When the status is `ACTIVE`, go on to the next step.

```
aws kendra describe-data-source \
 --id data source ID \
 --index-id index ID
```

6. Enter the following command to synchronize the data source.

```
aws kendra start-data-source-sync-job \
 --id data source ID \
 --index-id index ID
```

7. Amazon Kendra will index your data source. The amount of time that it takes depends on the number of documents. You can check the status of the sync job using the following command. When the status is `ACTIVE`, go on to the next step.

```
aws kendra describe-data-source \
 --id data source ID \
 --index-id index ID
```

8. Enter the following command to make a query.

```
aws kendra query \
 --index-id index ID \
 --query-text "search term"
```

   The results of the search are displayed in JSON format.

Getting started (AWS SDK for Python (Boto3))

The following program is an example of using Amazon Kendra in a Python program. The program performs the following actions:

1. Creates a new index using the [CreateIndex](#) operation.
2. Waits for index creation to complete. It uses the [DescribeIndex](#) operation to monitor the status of the index.
3. Once the index is active, it creates a data source using the [CreateDataSource](#) operation.
4. Waits for data source creation to complete. It uses the [DescribeDataSource](#) operation to monitor the status of the data source.
5. When the data source is active, it synchronizes the index with the contents of the data source using the [StartDataSourceSyncJob](#) operation.

```
import boto3
```

```
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Create an index.")

# Provide a name for the index
index_name = "python-getting-started-index"
# Provide an optional decription for the index
description = "Getting started index"
# Provide the IAM role ARN required for indexes
index_role_arn = "arn:aws:iam::${accountId}:role/KendraRoleForGettingStartedIndex"

try:
    index_response = kendra.create_index(
        Description = description,
        Name = index_name,
        RoleArn = index_role_arn
    )

    pprint.pprint(index_response)

    index_id = index_response["Id"]

    print("Wait for Amazon Kendra to create the index.")

    while True:
        # Get the details of the index, such as the status
        index_description = kendra.describe_index(
            Id = index_id
        )
        # When status is not CREATING quit.
        status = index_description["Status"]
        print(" Creating index. Status: "+status)
        time.sleep(60)
        if status != "CREATING":
            break

    print("Create an S3 data source.")

    # Provide a name for the data source
    data_source_name = "python-getting-started-data-source"
```

```
    # Provide an optional description for the data source
    data_source_description = "Getting started data source."
    # Provide the IAM role ARN required for data sources
    data_source_role_arn = "arn:aws:iam::${accountId}:role/
KendraRoleForGettingStartedDataSource"
    # Provide the data source connection information
    S3_bucket_name = "S3-bucket-name"
    data_source_type = "S3"
    # Configure the data source
    configuration = {"S3Configuration":
        {
            "BucketName": S3_bucket_name
        }
    }

    """
    If you connect to your data source using a template schema,
    configure the template schema
    configuration = {"TemplateConfiguration":
        {
            "Template": {JSON schema}
        }
    }
    """

    data_source_response = kendra.create_data_source(
        Name = data_source_name,
        Description = data_source_name,
        RoleArn = data_source_role_arn,
        Type = data_source_type,
        Configuration = configuration,
        IndexId = index_id
    )

    pprint.pprint(data_source_response)

    data_source_id = data_source_response["Id"]

    print("Wait for Amazon Kendra to create the data source.")

    while True:
        # Get the details of the data source, such as the status
        data_source_description = kendra.describe_data_source(
            Id = data_source_id,
```

```
            IndexId = index_id
        )
        # If status is not CREATING, then quit
        status = data_source_description["Status"]
        print(" Creating data source. Status: "+status)
        time.sleep(60)
        if status != "CREATING":
            break

    print("Synchronize the data source.")

    sync_response = kendra.start_data_source_sync_job(
        Id = data_source_id,
        IndexId = index_id
    )

    pprint.pprint(sync_response)

    print("Wait for the data source to sync with the index.")

    while True:

        jobs = kendra.list_data_source_sync_jobs(
            Id = data_source_id,
            IndexId = index_id
        )

        # For this example, there should be one job
        status = jobs["History"][0]["Status"]

        print(" Syncing data source. Status: "+status)
        if status != "SYNCING":
            break
        time.sleep(60)

except  ClientError as e:
        print("%s" % e)

print("Program ends.")
```

Getting started (AWS SDK for Java)

The following program is an example of using Amazon Kendra in a Java program. The program performs the following actions:

1. Creates a new index using the CreateIndex operation.
2. Waits for index creation to complete. It uses the DescribeIndex operation to monitor the status of the index.
3. Once the index is active, it creates a data source using the CreateDataSource operation.
4. Waits for data source creation to complete. It uses the DescribeDataSource operation to monitor the status of the data source.
5. When the data source is active, it synchronizes the index with the contents of the data source using the StartDataSourceSyncJob operation.

```
package com.amazonaws.kendra;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.CreateDataSourceRequest;
import software.amazon.awssdk.services.kendra.model.CreateDataSourceResponse;
import software.amazon.awssdk.services.kendra.model.CreateIndexRequest;
import software.amazon.awssdk.services.kendra.model.CreateIndexResponse;
import software.amazon.awssdk.services.kendra.model.DataSourceConfiguration;
import software.amazon.awssdk.services.kendra.model.DataSourceStatus;
import software.amazon.awssdk.services.kendra.model.DataSourceSyncJob;
import software.amazon.awssdk.services.kendra.model.DataSourceSyncJobStatus;
import software.amazon.awssdk.services.kendra.model.DataSourceType;
import software.amazon.awssdk.services.kendra.model.DescribeDataSourceRequest;
import software.amazon.awssdk.services.kendra.model.DescribeDataSourceResponse;
import software.amazon.awssdk.services.kendra.model.DescribeIndexRequest;
import software.amazon.awssdk.services.kendra.model.DescribeIndexResponse;
import software.amazon.awssdk.services.kendra.model.IndexStatus;
import software.amazon.awssdk.services.kendra.model.ListDataSourceSyncJobsRequest;
import software.amazon.awssdk.services.kendra.model.ListDataSourceSyncJobsResponse;
import software.amazon.awssdk.services.kendra.model.S3DataSourceConfiguration;
import software.amazon.awssdk.services.kendra.model.StartDataSourceSyncJobRequest;
import software.amazon.awssdk.services.kendra.model.StartDataSourceSyncJobResponse;


public class CreateIndexAndDataSourceExample {
```

```
    public static void main(String[] args) throws InterruptedException {
        System.out.println("Create an index");

        String indexDescription = "Getting started index for Kendra";
        String indexName = "java-getting-started-index";
        String indexRoleArn = "arn:aws:iam::<your AWS account ID>:role/<name of an IAM
 role>";

        System.out.println(String.format("Creating an index named %s", indexName));
        KendraClient kendra = KendraClient.builder().build();

        CreateIndexRequest createIndexRequest = CreateIndexRequest
            .builder()
            .description(indexDescription)
            .name(indexName)
            .roleArn(indexRoleArn)
            .build();
        CreateIndexResponse createIndexResponse =
 kendra.createIndex(createIndexRequest);
        System.out.println(String.format("Index response %s", createIndexResponse));

        String indexId = createIndexResponse.id();

        System.out.println(String.format("Waiting until the index with index ID %s is
 created", indexId));
        while (true) {
            DescribeIndexRequest describeIndexRequest =
 DescribeIndexRequest.builder().id(indexId).build();
            DescribeIndexResponse describeIndexResponse =
 kendra.describeIndex(describeIndexRequest);
            IndexStatus status = describeIndexResponse.status();
            if (status != IndexStatus.CREATING) {
                break;
            }

            TimeUnit.SECONDS.sleep(60);
        }

        System.out.println("Creating an S3 data source");
        String dataSourceName = "java-getting-started-data-source";
        String dataSourceDescription = "Getting started data source";
        String s3BucketName = "amzn-s3-demo-bucket";
```

```
        String dataSourceRoleArn = "arn:aws:iam::<your AWS account ID>:role/<name of an
 IAM role>";

        CreateDataSourceRequest createDataSourceRequest = CreateDataSourceRequest
            .builder()
            .indexId(indexId)
            .name(dataSourceName)
            .description(dataSourceDescription)
            .roleArn(dataSourceRoleArn)
            .type(DataSourceType.S3)
            .configuration(
                DataSourceConfiguration
                    .builder()
                    .s3Configuration(
                        S3DataSourceConfiguration
                            .builder()
                            .bucketName(s3BucketName)
                            .build()
                    ).build()
            ).build();

        CreateDataSourceResponse createDataSourceResponse =
 kendra.createDataSource(createDataSourceRequest);
        System.out.println(String.format("Response of creating data source: %s",
 createDataSourceResponse));

        String dataSourceId = createDataSourceResponse.id();
        System.out.println(String.format("Waiting for Kendra to create the data source
 %s", dataSourceId));
        DescribeDataSourceRequest describeDataSourceRequest = DescribeDataSourceRequest
            .builder()
            .indexId(indexId)
            .id(dataSourceId)
            .build();

        while (true) {
            DescribeDataSourceResponse describeDataSourceResponse =
 kendra.describeDataSource(describeDataSourceRequest);

            DataSourceStatus status = describeDataSourceResponse.status();
            System.out.println(String.format("Creating data source. Status: %s",
 status));
            if (status != DataSourceStatus.CREATING) {
                break;
```

```
            }

            TimeUnit.SECONDS.sleep(60);
        }

        System.out.println(String.format("Synchronize the data source %s", 
dataSourceId));
        StartDataSourceSyncJobRequest startDataSourceSyncJobRequest = 
StartDataSourceSyncJobRequest
            .builder()
            .indexId(indexId)
            .id(dataSourceId)
            .build();
        StartDataSourceSyncJobResponse startDataSourceSyncJobResponse = 
kendra.startDataSourceSyncJob(startDataSourceSyncJobRequest);
        System.out.println(String.format("Waiting for the data 
source to sync with the index %s for execution ID %s", indexId, 
startDataSourceSyncJobResponse.executionId()));

        // For this particular list, there should be just one job
        ListDataSourceSyncJobsRequest listDataSourceSyncJobsRequest = 
ListDataSourceSyncJobsRequest
            .builder()
            .indexId(indexId)
            .id(dataSourceId)
            .build();

        while (true) {
            ListDataSourceSyncJobsResponse listDataSourceSyncJobsResponse = 
kendra.listDataSourceSyncJobs(listDataSourceSyncJobsRequest);
            DataSourceSyncJob job = listDataSourceSyncJobsResponse.history().get(0);
            System.out.println(String.format("Syncing data source. Status: %s", 
job.status()));

            if (job.status() != DataSourceSyncJobStatus.SYNCING) {
                break;
            }

            TimeUnit.SECONDS.sleep(60);

        }

        System.out.println("Index setup is complete");
    }
```

```
}
```

Getting started with an Amazon S3 data source (console)

You can use the Amazon Kendra console to get started using an Amazon S3 bucket as a data store. When you use the console you specify all of the connection information you need to index the contents of the bucket. For more information, see Amazon S3.

Use the following procedure to create a basic S3 bucket data source using the default configuration. The procedure assumes that you created an index following the steps in step 1 of Getting started with the Amazon Kendra console.

To create an S3 bucket data source using the Amazon Kendra console

1. Sign into the AWS Management Console and open the Amazon Kendra console at https://console.aws.amazon.com/kendra/home.
2. From the list of indexes, choose the index that you want to add the data source to.
3. Choose **Add data sources**.
4. From the list of data source connectors, choose **Amazon S3**.
5. On the **Define attributes** page, give your data source a name and optionally a description. Leave the **Tags** field blank. Choose **Next** to continue.
6. In the **Enter the data source location** field, enter the name of the S3 bucket that contains your documents. You can enter the name directly, or you can browse for the name by choosing **Browse**. The bucket must be in the same Region as the index.
7. In **IAM role** choose **Create a new role** and then type a role name. For more information, see IAM roles for Amazon S3 data sources.
8. In the **Set sync run schedule** section, choose **Run on demand**.
9. Choose **Next** to continue.
10. On the **Review and create** page review the details of your S3 data source. If you want to make changes, choose the **Edit** button next to the item that you want to change. When you are satisfied with your choices, choose **Create** to create your S3 data source.

After you choose **Create**, Amazon Kendra starts creating the data source. It can take several minutes for the data source to be created. When it is finished, the status of the data source changes from **Creating** to **Active**.

After creating the data source, you need to sync the Amazon Kendra index with the data source. Choose **Sync now** to start the sync process. It can take several minutes to several hours to synchronize the data source, depending on the number and size of the documents.

Getting started with a MySQL database data source (console)

You can use the Amazon Kendra console to get started using a MySQL database as a data source. When you use the console you specify the connection information you need to index the contents of a MySQL database. For more information, see Using a database data source.

You first need to create a MySQL database, then you can create a data source for the database.

Use the following procedure to create a basic MySQL database. The procedure assumes that you have already created an index following step 1 of Getting started with the Amazon Kendra console.

To create a MySQL database

1. Sign in to the AWS Management Console and open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. From the navigation pane, choose **Subnet groups** and then choose **Create DB Subnet Group**.
3. Name the group and choose your Virtual Private Cloud (VPC). For more information on configuring a VPC, see Configuring Amazon Kendra to use a VPC.
4. Add your VPC's private subnets. Your private subnets are the ones that are not connected to your NAT. Choose **Create**.
5. From the navigation pane, choose **Databases** and then choose **Create database.**
6. Use the following parameters to create the database. Leave all of the other parameters at their defaults.

   - **Engine options**—MySQL
   - **Templates**—Free tier
   - **Credential Settings**—Enter and confirm a password
   - Under **Connectivity**, choose **Additional connectivity configuration**. Make the following choices.
     - **Subnet group**—Choose the subnet group that you created in step 4.
     - **VPC security group**—Choose the group that contains both inbound and outbound rules that you created in your VPC. For example, **`DataSourceSecurityGroup`**. For more information on configuring a VPC, see Configuring Amazon Kendra to use a VPC.

- Under **Additional configuration**, set the **Initial database name** to **`content`**.

7. Choose **Create database**.
8. From the list of databases, choose your new database. Make a note of the database endpoint.
9. After you create your database, you must create a table to hold your documents. Creating a table is outside the scope of these instructions. When you create your table, note the following:

   - Database name—**`content`**
   - Table name—**`documents`**
   - Columns—**`ID`**, **`Title`**, **`Body`**, and **`LastUpdate`**. You can include additional columns if you want.

Now that you have created your MySQL database, you can create a data source for the database.

To create a MySQL data source

1. Sign in to the AWS Management Console and open the Amazon Kendra console at https://console.aws.amazon.com/kendra/home.
2. From the navigation pane, choose **Indexes** and then choose your index.
3. Choose **Add data sources** and then choose **Amazon RDS**.
4. Type a name and description for the data source and then choose **Next**.
5. Choose **MySQL**.
6. Under **Connection access**, enter the following information:

   - **Endpoint**—The endpoint of the database that you created earlier.
   - **Port**—The port number for the database. For MySQL, the default is 3306.
   - **Type of authentication**—Choose **New**.
   - **New secret container name**—A name for the Secrets Manager container for the database credentials.
   - **Username**—The name of a user with administrative access to the database.
   - **Password**—The password for the user, and then choose **Save authentication**.
   - **Database name—`content`**.
   - **Table name—`documents`**.
   - **IAM role**—Choose **Create a new role**, and then type a name for the role.

7. In **Column configuration** enter the following:

   - **Document ID column name—`ID`**
   - **Document title column name—`Title`**
   - **Document data column name—Body**

8. In **Column change detection** enter the following:

   - **Change detecting columns—`LastUpdate`**

9. In **Configure VPC & security group** provide the following:

   - In **Virtual Private Cloud (VPC)**, choose your VPC.
   - In **Subnets**, choose the private subnets that you created in your VPC.
   - In **VPC security groups**, choose the security group that contains both inbound and outbound rules that you created in your VPC for MySQL databases. For example, **`DataSourceSecurityGroup`**.

10. In **Set sync run schedule**, choose **Run on demand** and then choose **Next**.
11. In **Data source field mapping**, choose **Next**.
12. Review the configuration of your data source to make sure that it is correct. When you're satisfied that everything is correct, choose **Create**.

Getting started with an AWS IAM Identity Center identity source (console)

An AWS IAM Identity Center identity source contains information on your users and groups. This is useful for setting up user context filtering, where Amazon Kendra filters search results for different users based on the user or their group's access to documents.

To create an IAM Identity Center identity source, you must activate IAM Identity Center and create an organization in AWS Organizations. When you activate IAM Identity Center and create an organization for the first time, it automatically defaults to the Identity Center directory as the identity source. You can change to Active Directory (Amazon managed or self-managed) or an external identity provider as your identity source. You must follow the correct guidance for this — see Changing your IAM Identity Center identity source. You can have only one identity source per organization.

In order for your users and groups to be assigned different levels of access to documents, you need to include your users and groups in your access control list when you ingest documents into your index. This allows your users and groups to search for documents in Amazon Kendra in accordance with their level of access. When you issue a query, the user ID needs to be an exact match of the user name in IAM Identity Center.

You must also grant the required permissions to use IAM Identity Center with Amazon Kendra. For more information, see IAM roles for IAM Identity Center.

To set up an IAM Identity Center identity source

1. Open the IAM Identity Center console.
2. Choose **Enable IAM Identity Center**, and then choose **Create AWS organization**.

   Identity Center directory is created by default, and an email is sent to you to verify the email address associated with the organization.
3. To add a group to your AWS organization, in the navigation pane, choose **Groups**.
4. On the **Groups page**, choose **Create group** and enter a group name and description in the dialog box. Choose **Create**.
5. To add a user to your Organizations, in the navigation pane, choose **Users**.
6. On the **Users** page, choose **Add user**. Under **User details**, specify all required fields. For **Password**, choose **Send an email to the user**. Choose **Next**.
7. To add a user to a group, choose **Groups** and select a group.
8. On the **Details** page, under **Group members**, choose **Add user**.
9. On the **Add users to group** page, select the user you want to add as a member of the group. You can select multiple users to add to a group.
10. To sync your list of users and groups with IAM Identity Center, change your identity source to Active Directory or External identity provider.

    Identity Center directory is the default identity source and requires you to manually add your users and groups using this source if you do not have your own list managed by a provider. To change your identity source, you must follow the correct guidance for this—see Changing your IAM Identity Center identity source.

Note

If using Active Directory or an external identity provider as your identity source, you must map the email addresses of your users to IAM Identity Center user names when you specify the System for Cross-domain Identity Management (SCIM) protocol. For more information, see the IAM Identity Center guide on SCIM for enabling IAM Identity Center.

Once you have set up your IAM Identity Center identity source, you can activate this in the console when you create or edit your index. Go to **User access control** in your index settings and edit your settings to allow fetching user-group information from IAM Identity Center.

You can also activate IAM Identity Center using the UserGroupResolutionConfiguration object. You provide the `UserGroupResolutionMode` as `AWS_SSO` and create an IAM role that gives permission to call `sso:ListDirectoryAssociations`, `sso-directory:SearchUsers`, `sso-directory:ListGroupsForUser`, `sso-directory:DescribeGroups`.

Warning

Amazon Kendra currently does not support using `UserGroupResolutionConfiguration` with an AWS organization member account for your IAM Identity Center identity source. You must create your index in the management account for the organization in order to use `UserGroupResolutionConfiguration`.

The following is an overview of how to set up a data source with `UserGroupResolutionConfiguration` and user access control to filter search results on user context. This assumes you have already created an index and an IAM role for indexes. You create an index and provide the IAM role using the CreateIndex API.

Setting up a data source with `UserGroupResolutionConfiguration` and user context filtering

1. Create an IAM role that gives permission to access your IAM Identity Center identity source.
2. Configure `UserGroupResolutionConfiguration` by setting the mode to `AWS_SSO` and call UpdateIndex to update your index to use IAM Identity Center.
3. If you want to use token-based user access control to filter search results on user context, set UserContextPolicy to `USER_TOKEN` when you call `UpdateIndex`. Otherwise, Amazon

Kendra crawls the access control list for each of your documents for most data source connectors. You can also filter search results on user context in the Query API by providing user and group information in `UserContext`. You can also map users to their groups using PutPrincipalMapping so that you only need to provide the user ID when you issue the query.

4. Create an IAM role that gives permission to access your data source.
5. Configure your data source. You must provide the required connection information to connect to your data source.
6. Create a data source using the CreateDataSource API. Provide the `DataSourceConfiguration` object, which includes `TemplateConfiguration`, the ID of your index, the IAM role for your data source, the data source type, and give your data source a name. You can also update your data source.

Changing your IAM Identity Center identity source

Warning

Changing your identity source in IAM Identity Center **Settings** might affect the preservation of user and group information. To do this safely, it is recommended you review Considerations for changing your identity source. When you change your identity source, a new identity source ID is generated. Check you are using the correct ID before you set the mode to `AWS_SSO` in UserGroupResolutionConfiguration.

To change your IAM Identity Center identity source

1. Open the IAM Identity Center> console.
2. Choose **Settings**.
3. On the **Settings** page, under **Identity source**, choose **Change**.
4. On the **Change identity source** page, select your preferred identity source, and then choose **Next**.

Creating an index

You can create an index using the console, or by calling the CreateIndex API. You can use the AWS Command Line Interface (AWS CLI) or SDK with the API. After you created your index, you can add documents directly to it or from a data source.

To create an index, you must provide the Amazon Resource Name (ARN) of an AWS Identity and Access Management (IAM) role for indexes to access CloudWatch. For more information, see IAM roles for indexes.

The following tabs provide a procedure for creating an index by using the AWS Management Console, and code examples for using the AWS CLI, and Python and Java SDKs.

Console

To create an index

1. Sign in to the AWS Management Console and open the Amazon Kendra console at https://console.aws.amazon.com/kendra/.
2. Select **Create index** in the **Indexes** section.
3. In **Specify index details**, give your index a name and a description.
4. In **IAM role** provide an IAM role. To find a role, choose from roles in your account that contain the word "kendra" or enter the name of another role. For more information about the permissions that the role requires, see IAM roles for indexes.
5. Choose **Next**.
6. On the **Configure user access control** page, choose **Next**. You can update your index to use tokens for access control after you create an index. For more information, see Controlling access to documents.
7. On the **Provisioning details** page, choose **Create**.
8. It might take some time for the index to create. Check the list of indexes to watch the progress of creating your index. When the status of the index is `ACTIVE`, your index is ready to use.

AWS CLI

To create an index

1. Use the following command to create an index. The `role-arn` must be the Amazon Resource Name (ARN) of an IAM role that can run Amazon Kendra actions. For more information, see IAM roles.

   The command is formatted for Linux and macOS. If you are using Windows, replace the Unix line continuation character (\) with a caret (^).

```
aws kendra create-index \
 --name index name \
 --description "index description" \
 --role-arn arn:aws:iam::account ID:role/role name
```

2. It might take some time for the index to create. To check the state of your index, use the index ID returned by `create-index` with the following command. When the status of the index is `ACTIVE`, your index is ready to use.

```
aws kendra describe-index \
 --index-id index ID
```

Python

To create an index

- Provide values for the following variables in the code example that follows:
  - `description`—A description of the index that you're creating. This is optional.
  - `index_name`—The name of the index that you're creating.
  - `role_arn`—The Amazon Resource Name (ARN) of a role that can run Amazon Kendra APIs. For more information, see IAM roles.

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time
```

```
kendra = boto3.client("kendra")

print("Create an index.")

# Provide a name for the index
index_name = "index-name"
# Provide an optional description for the index
description = "index description"
# Provide the IAM role ARN required for indexes
role_arn = "arn:aws:iam::${account id}:role/${role name}"

try:
    index_response = kendra.create_index(
        Name = index_name,
        Description = description,
        RoleArn = role_arn
    )

    pprint.pprint(index_response)

    index_id = index_response["Id"]

    print("Wait for Amazon Kendra to create the index.")

    while True:
        # Get the details of the index, such as the status
        index_description = kendra.describe_index(
            Id = index_id
        )
        # If status is not CREATING, then quit
        status = index_description["Status"]
        print(" Creating index. Status: "+status)
        if status != "CREATING":
            break
        time.sleep(60)

except  ClientError as e:
        print("%s" % e)

print("Program ends.")
```

Java

To create an index

- Provide values for the following variables in the code example that follows:
  - `description`—A description of the index that you're creating. This is optional.
  - `index_name`—The name of the index that you're creating.
  - `role_arn`—The Amazon Resource Name (ARN) of a role that can run Amazon Kendra APIs. For more information, see IAM roles.

```
package com.amazonaws.kendra;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.CreateIndexRequest;
import software.amazon.awssdk.services.kendra.model.CreateIndexResponse;
import software.amazon.awssdk.services.kendra.model.DescribeIndexRequest;
import software.amazon.awssdk.services.kendra.model.DescribeIndexResponse;
import software.amazon.awssdk.services.kendra.model.IndexStatus;


public class CreateIndexExample {

    public static void main(String[] args) throws InterruptedException {

        String indexDescription = "Getting started index for Kendra";
        String indexName = "java-getting-started-index";
        String indexRoleArn = "arn:aws:iam::<your AWS account ID>:role/
KendraRoleForGettingStartedIndex";

        System.out.println(String.format("Creating an index named %s",
 indexName));
        CreateIndexRequest createIndexRequest = CreateIndexRequest
            .builder()
            .description(indexDescription)
            .name(indexName)
            .roleArn(indexRoleArn)
            .build();
        KendraClient kendra = KendraClient.builder().build();
```

```
        CreateIndexResponse createIndexResponse =
 kendra.createIndex(createIndexRequest);
        System.out.println(String.format("Index response %s",
 createIndexResponse));

        String indexId = createIndexResponse.id();

        System.out.println(String.format("Waiting until the index with ID %s is
 created.", indexId));
        while (true) {
            DescribeIndexRequest describeIndexRequest =
 DescribeIndexRequest.builder().id(indexId).build();
            DescribeIndexResponse describeIndexResponse =
 kendra.describeIndex(describeIndexRequest);
            IndexStatus status = describeIndexResponse.status();
            if (status != IndexStatus.CREATING) {
                break;
            }

            TimeUnit.SECONDS.sleep(60);
        }

        System.out.println("Index creation is complete.");
    }
}
```

After you created your index, you add documents to it. You can add them directly or create a data source that updates your index on a regular schedule.

Topics

Adding documents directly to an index with batch upload

You can add documents directly to an index using the BatchPutDocument API. You can't add documents directly using the console. If you use the console, you connect to a data source to add documents to your index. Documents can be added from an S3 bucket or supplied as binary data. For a list of document types supported by Amazon Kendra see Types of documents.

Adding documents to an index using `BatchPutDocument` is an asynchronous operation. After you call the `BatchPutDocument` API, you use the BatchGetDocumentStatus API to monitor the progress of indexing your documents. When you call the `BatchGetDocumentStatus` API with a list of document IDs, it returns the status of the document. When the status of the document is `INDEXED` or `FAILED`, processing of the document is complete. When the status is `FAILED`, the `BatchGetDocumentStatus` API returns the reason that the document couldn't be indexed.

If you want to alter your content and document metadata fields or attributes during the document ingestion process, see Amazon Kendra Custom Document Enrichment. If you want to use a custom data source, each document you submit using the `BatchPutDocument` API requires a data source ID and execution ID as attributes or fields. For more information, see Required attributes for custom data sources.

> **Note**
>
> Each document ID must be unique per index. You cannot create a data source to index your documents with their unique IDs and then use the `BatchPutDocument` API to index the same documents, or vice versa. You can delete a data source and then use the `BatchPutDocument` API to index the same documents, or vice versa. Using the `BatchPutDocument` and `BatchDeleteDocument` APIs in combination with an Amazon Kendra data source connector for the same set of documents could cause inconsistencies with your data. Instead, we recommend using the Amazon Kendra custom data source connector.

The following developer guide documents show how to add documents directly to an index.

Topics

- Adding documents with the BatchPutDocument API
- Adding documents from an S3 bucket

Adding documents with the BatchPutDocument API

The following example adds a blob of text to an index by calling [BatchPutDocument](). You can use the `BatchPutDocument` API to add documents directly to your index. For a list of document types supported by Amazon Kendra see [Types of documents]().

For an example of creating an index using the AWS CLI and SDKs, see [Creating an index](). To set up the CLI and SDKs, see [Setting up Amazon Kendra]().

> **Note**
>
> Files added to the index must be in a UTF-8 encoded byte stream.

In the following examples, UTF-8 encoded text is added to the index.

CLI

In the AWS Command Line Interface, use the following command. The command is formatted for Linux and macOS. If you are using Windows, replace the Unix line continuation character (\) with a caret (^).

```
aws kendra batch-put-document \
   --index-id index-id \
   --documents '{"Id":"doc-id-1", "Blob":"Amazon.com is an online retailer.",
 "ContentType":"PLAIN_TEXT", "Title":"Information about Amazon.com"}'
```

Python

```
import boto3

kendra = boto3.client("kendra")

# Provide the index ID
index_id = "index-id"

# Provide the title and text
title = "Information about Amazon.com"
text = "Amazon.com is an online retailer."

document = {
```

```
    "Id": "1",
    "Blob": text,
    "ContentType": "PLAIN_TEXT",
    "Title": title
}

documents = [
    document
]

result = kendra.batch_put_document(
    IndexId = index_id,
    Documents = documents
)

print(result)
```

Java

```
package com.amazonaws.kendra;



import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.BatchPutDocumentRequest;
import software.amazon.awssdk.services.kendra.model.BatchPutDocumentResponse;
import software.amazon.awssdk.services.kendra.model.ContentType;
import software.amazon.awssdk.services.kendra.model.Document;

public class AddDocumentsViaAPIExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        String indexId = "yourIndexId";

        Document testDoc = Document
            .builder()
            .title("The title of your document")
            .id("a_doc_id")
            .blob(SdkBytes.fromUtf8String("your text content"))
            .contentType(ContentType.PLAIN_TEXT)
            .build();

```

```
        BatchPutDocumentRequest batchPutDocumentRequest = BatchPutDocumentRequest
            .builder()
            .indexId(indexId)
            .documents(testDoc)
            .build();

        BatchPutDocumentResponse result =
 kendra.batchPutDocument(batchPutDocumentRequest);

        System.out.println(String.format("BatchPutDocument Result: %s", result));
    }
}
```

Adding documents from an S3 bucket

You can add documents directly to your index from an Amazon S3 bucket using the BatchPutDocument API. You can add up to 10 documents in the same call. When you use an S3 bucket, you must provide an IAM role with permission to access the bucket that contains your documents. You specify the role in the `RoleArn` parameter.

Using the BatchPutDocument API to add documents from an Amazon S3 bucket is a one-time operation. To keep an index synchronized with the contents of a bucket, create an Amazon S3 data source. For more information, see Amazon S3 data source.

For an example of creating an index using the AWS CLI and SDKs, see Creating an index. To set up the CLI and SDKs, see Setting up Amazon Kendra. For information on creating an S3 bucket, see Amazon Simple Storage Service documentation.

In the following example, two Microsoft Word documents are added to the index using the `BatchPutDocument` API.

Python

```
import boto3

kendra = boto3.client("kendra")

# Provide the index ID
index_id = "index-id"
# Provide the IAM role ARN required to index documents in an S3 bucket
role_arn = "arn:aws:iam::${acccountID}:policy/${roleName}"
```

```
doc1_s3_file_data = {
    "Bucket": "bucket-name",
    "Key": "document1.docx"
}

doc1_document = {
    "S3Path": doc1_s3_file_data,
    "Title": "Document 1 title",
    "Id": "doc_1"
}

doc2_s3_file_data = {
    "Bucket": "bucket-name",
    "Key": "document2.docx"
}

doc2_document = {
    "S3Path": doc2_s3_file_data,
    "Title": "Document 2 title",
    "Id": "doc_2"
}

documents = [
    doc1_document,
    doc2_document
]

result = kendra.batch_put_document(
    Documents = documents,
    IndexId = index_id,
    RoleArn = role_arn
)

print(result)
```

Java

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.BatchPutDocumentRequest;
import software.amazon.awssdk.services.kendra.model.BatchPutDocumentResponse;
```

```
import software.amazon.awssdk.services.kendra.model.Document;
import software.amazon.awssdk.services.kendra.model.S3Path;

public class AddFilesFromS3Example {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        String indexId = "yourIndexId";
        String roleArn = "yourIndexRoleArn";

        Document pollyDoc = Document
            .builder()
            .s3Path(
                S3Path.builder()
                .bucket("amzn-s3-demo-bucket")
                .key("What is Amazon Polly.docx")
                .build())
            .title("What is Amazon Polly")
            .id("polly_doc_1")
            .build();

        Document rekognitionDoc = Document
            .builder()
            .s3Path(
                S3Path.builder()
                .bucket("amzn-s3-demo-bucket")
                .key("What is Amazon Rekognition.docx")
                .build())
            .title("What is Amazon rekognition")
            .id("rekognition_doc_1")
            .build();

        BatchPutDocumentRequest batchPutDocumentRequest = BatchPutDocumentRequest
            .builder()
            .indexId(indexId)
            .roleArn(roleArn)
            .documents(pollyDoc, rekognitionDoc)
            .build();

        BatchPutDocumentResponse result =
 kendra.batchPutDocument(batchPutDocumentRequest);

        System.out.println(String.format("BatchPutDocument result: %s", result));
    }
```

```
}
```

Adding frequently asked questions (FAQs) to an index

Note

Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

You can add frequently asked questions (FAQs) directly to your index using the console or the CreateFaq API. Adding FAQs to an index is an asynchronous operation. You put the data for the FAQ in a file that you store in an Amazon Simple Storage Service bucket. You can use CSV or JSON files as input for your FAQ:

- Basic CSV—A CSV file where each row contains a question, answer, and an optional source URI.
- Custom CSV—A CSV file that contains questions, answers, and headers for custom fields/attributes that you can use to facet, display, or sort FAQ responses. You can also define access control fields to limit the FAQ response to certain users and groups that are allowed to see the FAQ response.
- JSON—A JSON file that contains questions, answers, and custom fields/attributes that you can use to facet, display, or sort FAQ responses. You can also define access control fields to limit the FAQ response to certain users and groups that are allowed to see the FAQ response.

For example, the following is a basic CSV file that provides answers to questions about free clinics in Spokane, Washington USA and Mountain View, Missouri, USA.

```
How many free clinics are in Spokane WA?, 13
How many free clinics are there in Mountain View Missouri?, 7
```

Note

The FAQ file must be a UTF-8-encoded file.

Topics

Creating index fields for an FAQ file

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

When you use a custom CSV or JSON file for input, you can declare custom fields for your FAQ questions. For example, you can create a custom field that assigns each FAQ question a business department. When the FAQ is returned in a response, you can use the department as a facet to narrow the search to "HR" or "Finance" only, for example.

A custom field must map to an index field. In the console, you use the **Facet definition** page to create an index field. When using the API, you must first create an index field using the UpdateIndex API.

The field/attribute type in the FAQ file must match the type of the associated index field. For example, the "Department" field is a `STRING_LIST` type field. So, you must provide values for the department field as a string list in your FAQ file. You can check the type of index fields using the **Facet definition** page in the console or by using the DescribeIndex API.

When you create an index field that maps to a custom attribute, you can mark it displayable, facetable, or sortable. You can't make a custom attribute searchable.

In addition to the custom attributes, you can also use the Amazon Kendra reserved or common fields in a custom CSV or JSON file. For more information, see Document attributes or fields.

Basic CSV file

Use a basic CSV file when you want to use a simple structure for your FAQs. In a basic CSV file, each row has two or three fields: a question, an answer, and an optional source URI that points to a document with more information.

The contents of the file must follow the [RFC 4180 Common Format and MIME Type for Comma-Separated Values (CSV) Files].

The following is a FAQ file in the basic CSV format.

```
How many free clinics are in Spokane WA?, 13, https://s3.region.company.com/bucket-
name/directory/faq.csv
How many free clinics are there in Mountain View Missouri?, 7, https://
s3.region.company.com/bucket-name/directory/faq.csv
```

Custom CSV file

Use a custom CSV file when you want to add custom fields/attributes to your FAQ questions. For a custom CSV file, you use a header row in your CSV file to define the additional attributes.

The CSV file must contain the following two required fields:

- `_question`—The frequently asked question
- `_answer`—The answer to the frequently asked question

Your custom CSV file can contain both Amazon Kendra reserved fields (except `_faq_id`, `_data_source_id`, `_document_title`, and `_file_type`) and any custom fields.

The following is an example of a custom CSV file.

```
_question,_answer,_last_updated_at,custom_string
How many free clinics are in Spokane WA?, 13, 2012-03-25T12:30:10+01:00, Note: Some
 free clinics require you to meet certain criteria in order to use their services
How many free clinics are there in Mountain View Missouri?, 7,
 2012-03-25T12:30:10+01:00, Note: Some free clinics require you to meet certain
 criteria in order to use their services
```

The contents of the custom file must follow the [RFC 4180 Common Format and MIME Type for Comma-Separated Values (CSV) Files].

The following lists the types of custom fields:

- Date—ISO 8601-encoded date and time values.

  For example, 2012-03-25T12:30:10+01:00 is the ISO 8601 date-time format for March 25, 2012, at 12:30PM (plus 10 seconds) in the Central European Time time zone.
- Long—Numbers, such as `1234`.
- String—String values. If your string contains commas, enclose the entire value in double quotation marks (") (for example, `"custom attribute, and more"`).
- String list—A list of string values. List the values in a comma-separated list that's enclosed in quotation marks (") (for example, `"item1, item2, item3"`). If the list contains only a single entry, you can omit the quotation marks (for example, `item1`).

A custom CSV file can contain user access control fields. You can use these fields to limit access to the FAQ to certain users and groups. To filter on user context, the user must provide user and group information in the query. Otherwise, all relevant FAQs are returned. For more information, see User context filtering.

There following lists the user context filters for FAQs:

- `_acl_user_allow`—Users in the allow list can see the FAQ in the query response. The FAQ isn't returned to other users.
- `_acl_user_deny`—Users in the deny list can't see the FAQ in the query response. The FAQ is returned to all other users when it's relevant to the query.
- `_acl_group_allow`—Users that are members of an allowed group can see the FAQ in the query response. The FAQ isn't returned to users that are members of another group.
- `_acl_group_deny`—Users that are members of a denied group can't see the FAQ in the query response. The FAQ is returned to other groups when it's relevant to the query.

Provide the values for the allow and deny lists in comma-separated lists enclosed in quotation marks (for example, `"user1,user2,user3"`). You can include a user or a group in either an allow list or a deny list, but not both where the same user is individually allowed but also group denied. If you include a user or group in both, you receive an error.

The following is an example of a custom CSV file with user context information.

```
_question, _answer, _acl_user_allow, _acl_user_deny, _acl_group_allow, _acl_group_deny
```

```
How many free clinics are in Spokane WA?, 13, "userID6201,userID7552",
 "userID1001,userID2020", groupBasicPlusRate, groupPremiumRate
```

JSON file

You can use a JSON file to provide questions, answers, and fields for your index. You can add any of the Amazon Kendra reserved fields or custom fields to the FAQ.

The following is the schema for the JSON file.

```
{
    "SchemaVersion": 1,
    "FaqDocuments": [
        {
            "Question": string,
            "Answer": string,
            "Attributes": {
                string: object
                additional attributes
            },
            "AccessControlList": [
               {
                   "Name": string,
                   "Type": enum( "GROUP" | "USER" ),
                   "Access": enum( "ALLOW" | "DENY" )
               },
               additional user context
            ]
        },
        additional FAQ documents
    ]
}
```

The following example JSON file shows two FAQ documents. One of the documents has the required question and answer only. The other document also includes additional field and user context or access control information.

```
{
    "SchemaVersion": 1,
    "FaqDocuments": [
        {
```

```
            "Question": "How many free clinics are in Spokane WA?",
            "Answer": "13"
        },
        {
            "Question": "How many free clinics are there in Mountain View Missouri?",
            "Answer": "7",
            "Attributes": {
                "_source_uri": "https://s3.region.company.com/bucket-name/directory/
faq.csv",
                "_category": "Charitable Clinics"
            },
            "AccessControlList": [
               {
                   "Name": "user@amazon.com",
                   "Type": "USER",
                   "Access": "ALLOW"
               },
               {
                   "Name": "Admin",
                   "Type": "GROUP",
                   "Access": "ALLOW"
               }
            ]
        }
    ]
}
```

The following lists the types of custom fields:

- Date—A JSON string value with ISO 8601-encoded date and time values. For example, 2012-03-25T12:30:10+01:00 is the ISO 8601 date-time format for March 25, 2012, at 12:30PM (plus 10 seconds) in the Central European Time time zone.
- Long—A JSON number value, such as `1234`.
- String—A JSON string value (for example, `"custom attribute"`).
- String list—A JSON array of string values (for example, `["item1,item2,item3"]`).

A JSON file can contain user access control fields. You can use these fields to limit access to the FAQ to certain users and groups. To filter on user context, the user must provide user and group information in the query. Otherwise, all relevant FAQs are returned. For more information, see User context filtering.

You can include a user or a group in either an allow list or a deny list, but not both where the same user is individually allowed but also group denied. If you include a user or group in both, you receive an error.

The following is an example of including user access control to a JSON FAQ.

```
"AccessControlList": [
                {
                    "Name": "group or user name",
                    "Type": "GROUP | USER",
                    "Access": "ALLOW | DENY"
                },
                additional user context
            ]
```

Using your FAQ file

After you store your FAQ input file in an S3 bucket, you use the console or the `CreateFaq` API to put the questions and answers into your index. If you want to update a FAQ, delete the FAQ and create it again. You use the `DeleteFaq` API to delete a FAQ.

You must provide an IAM role that has access to the S3 bucket that contains your source files. You specify the role in the console or in the `RoleArn` parameter. The following is an example of adding a FAQ file to an index.

Python

```
import boto3

kendra = boto3.client("kendra")

# Provide the index ID
index_id = "index-id"
# Provide the IAM role ARN required to index documents in an S3 bucket
role_arn = "arn:aws:iam::${accountId}:role/${roleName}"

# Provide the S3 bucket path information to the FAQ file
faq_path = {
    "Bucket": "bucket-name",
    "Key": "FreeClinicsUSA.csv"
}
```

```
response = kendra.create_faq(
    S3Path =  faq_path,
    Name = "FreeClinicsUSA",
    IndexId = index_id,
    RoleArn = role_arn
)

print(response)
```

Java

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.CreateFaqRequest;
import software.amazon.awssdk.services.kendra.model.CreateFaqResponse;
import software.amazon.awssdk.services.kendra.model.S3Path;

public class AddFaqExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        String indexId = "yourIndexId";
        String roleArn = "your role for accessing S3 files";

        CreateFaqRequest createFaqRequest = CreateFaqRequest
            .builder()
            .indexId(indexId)
            .name("FreeClinicsUSA")
            .roleArn(roleArn)
            .s3Path(
                S3Path
                    .builder()
                    .bucket("amzn-s3-demo-bucket")
                    .key("FreeClinicsUSA.csv")
                    .build())
            .build();

        CreateFaqResponse response = kendra.createFaq(createFaqRequest);

        System.out.println(String.format("The result of creating FAQ: %s",
 response));
```

```
    }
}
```

FAQ files in languages other than English

You can index a FAQ in a supported language. Amazon Kendra indexes FAQs in English by default if you don't specify a language. You specify the language code when you call the CreateFaq operation or you can include the language code for a FAQ in the FAQ metadata as a field. If a FAQ doesn't have a language code in its metadata specified in a metadata field, the FAQ is indexed using the language code specified when you call the `CreateFAQ` operation. To index a FAQ document in a supported language in the console, go to **FAQs** and select **Add FAQ**. You choose a language from the dropdown **Language**.

Creating custom document fields

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Searching indexes.

You can create custom attributes or fields for your documents in your Amazon Kendra index. For example, you can create a custom field or attribute called "Department" with the values of "HR", "Sales", and "Manufacturing". If you map these custom fields or attributes to your Amazon Kendra index, you can use them to filter the search results to include documents by the "HR" department attribute, for example.

Before you can use a custom field or attribute, you must first create the field in the index. Use the console to edit the data source field mappings to add a custom field or use the UpdateIndex API to create the index field. You cannot change the field data type once you have created the field.

For most data sources, you map fields in the external data source to the corresponding fields in Amazon Kendra. For more information, see Mapping data source fields. For S3 data sources, you can create custom fields or attributes using a JSON metadata file.

You can create up to 500 custom fields or attributes.

You can also use Amazon Kendra reserved or common fields. For more information, see Document attributes or fields.

Topics

- Updating custom document fields

Updating custom document fields

With the `UpdateIndex` API, you add custom fields or attributes using the `DocumentMetadataConfigurationUpdates` parameter.

The following JSON example uses `DocumentMetadataConfigurationUpdates` to add a field called "Department" to the index.

```
"DocumentmetadataConfigurationUpdates": [
   {
       "Name": "Department",
       "Type": "STRING_VALUE"
   }
]
```

The following sections include examples for adding custom attributes or fields using the BatchPutDocument and for an Amazon S3 data source.

Topics

- Adding custom attributes or fields with the BatchPutDocument API
- Adding custom attributes or fields to an Amazon S3 data source

Adding custom attributes or fields with the BatchPutDocument API

When you use the BatchPutDocument API to add a document to your index, you specify custom fields or attributes as part of `Attributes`. You can add multiple fields or attributes when you call the API. You can create up to 500 custom fields or attributes. The following example is a custom field or attribute that adds "Department" to a document.

```
"Attributes":
    {
        "Department": "HR",
        "_category": "Vacation policy"
```

```
    }
```

Adding custom attributes or fields to an Amazon S3 data source

When you use an S3 bucket as a data source for your index, you add metadata to the documents with companion metadata files. You place the metadata JSON files in a directory structure that is parallel to your documents. For more information, see S3 document metadata.

You specify custom fields or attributes in the `Attributes` JSON structure. You can create up to 500 custom fields or attributes. For example, the following example uses `Attributes` to define three custom fields or attributes and one reserved field.

```
"Attributes": {
        "brand": "Amazon Basics",
        "price": 1595,
        "_category": "sports",
        "subcategories": ["outdoors", "electronics"]
    }
```

The following steps walk you through adding custom attributes to an Amazon S3 data source.

Topics

Step 1: Create a Amazon Kendra index

Follow the steps in Creating an index to create your Amazon Kendra index.

Step 2: Update index to add custom document fields

After creating an index, you add fields to it. The following procedure shows how to add fields to an index using the console and the CLI.

Console

To create index fields

1. Make sure you've created an index.

2. Then, from the left navigation menu, from **Data management**, choose **Facet definition**.
3. In **Index field settings guide**, from **Index fields**, choose **Add field** to add custom fields.
4. In the **Add index field** dialog box, do the following:

   - **Field name –** Add a field name.
   - **Data type –** Select data type, whether **String**, **String list**, or **Date**.
   - **Usage types –** Select usage types, whether **Facetable**, **Searchable**, **Displayable**, and **Sortable**.

     Then, select **Add**.

   Repeat the last step for any other fields you want to map.

CLI

```
aws kendra update-index  \
--region $region \
--endpoint-url $endpoint \
--application-id $applicationId \
--index-id $indexId  \
--document-metadata-configuration-updates \
"[
    {
        "Name": "string",
        "Type": "STRING_VALUE"|"STRING_LIST_VALUE"|"LONG_VALUE"|"DATE_VALUE",
        "Relevance": {
            "Freshness": true|false,
            "Importance": integer,
            "Duration": "string",
            "RankOrder": "ASCENDING"|"DESCENDING",
            "ValueImportanceMap": {"string": integer
            ...}
    },
    "Search": {
        "Facetable": true|false,
        "Searchable": true|false,
        "Displayable": true|false,
        "Sortable": true|false
        }
    }
```

```
...
]"
```

Step 3: Create an Amazon S3 data source and map data source fields to custom attributes

To create an Amazon S3 data source and map fields to it, follow the instructions in Amazon S3.

If you're using the API, use the `fieldMappings` attribute under `configuration` when you use the CreateDataSource API.

For an overview of how data source fields are mapped, see Mapping data source fields.

Controlling user access to documents with tokens

Note

Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

Important

Amazon Kendra GenAI Enterprise Edition indices don't support token-based user access control.

You can control which users or groups can access certain documents in your index or see certain documents in their search results. This is called user context filtering. It is a kind of personalized search with the benefit of controlling access to documents. For example, not all teams that search the company portal for information should access top-secret company documents, nor are these documents relevant to all users. Only specific users or groups of teams given access to top-secret documents should see these documents in their search results.

Amazon Kendra Enterprise and Developer indices support token-based user access control using the following token types:

- Open ID
- JWT with a shared secret

- JWT with a public key
- JSON

Amazon Kendra can be used to deliver secure enterprise search for your retrieval and search applications. During query and retrieval, Amazon Kendra filters search results based on `AttributeFilters` and `UserContext` provided in the request. Amazon Kendra reads document access control lists (ACLs) collected by the its connectors during crawl and ingestion. The retrieval and search results return URLs pointing back to the original document repositories plus short excerpts. Access to the full document is still enforced by the original repository.

Topics

Using OpenID

To configure an Amazon Kendra index to use an OpenID token for access control, you need the JWKS (JSON Web Key Set) URL from the OpenID provider. In most cases the JWKS URL is in the following format (if they're following openId discovery) `https://domain-name/.well_known/jwks.json`.

The following examples show how to use an OpenID token for user access control when you create an index.

Console

1. Choose **Create index** to start creating a new index.
2. On the **Specify index details** page, give your index a name and a description.
3. For **IAM role**, select a role or select **Create a new role** to and specify a role name to create a new role. The IAM role will have the prefix "AmazonKendra-".
4. Leave all of the other fields at their defaults. Choose **Next**.
5. In the **Configure user access control** page, under **Access control settings**, choose **Yes** to use tokens for access control.

6. Under **Token configuration**, select **OpenID** as the **Token type**.
7. Specify a **Signing key URL**. The URL should point to a set of JSON web keys.
8. *Optional* Under **Advanced configuration**:
   a. Specify a **Username** to use in the ACL check.
   b. Specify one or more **Groups** to use in the ACL check.
   c. Specify the **Issuer** that will validate the token issuer.
   d. Specify the **Client Id(s)**. You must specify a regular expression that match the audience in the JWT.
9. In the **Provisioning details** page, choose **Developer edition**.
10. Choose **Create** to create your index.
11. Wait for your index to be created. Amazon Kendra provisions the hardware for your index. This operation can take some time.

CLI

To create an index with the AWS CLI using a JSON input file, first create a JSON file with your desired parameters:

```
{
    "Name": "user-context",
    "Edition": "ENTERPRISE_EDITION",
    "RoleArn": "arn:aws:iam::account-id:role:/my-role",
    "UserTokenConfigurations": [
        {
            "JwtTokenTypeConfiguration": {
                "KeyLocation": "URL",
                "Issuer": "optional: specify the issuer url",
                "ClaimRegex": "optional: regex to validate claims in the token",
                "UserNameAttributeField": "optional: user",
                "GroupAttributeField": "optional: group",
                "URL": "https://example.com/.well-known/jwks.json"
            }
        }
    ],
    "UserContextPolicy": "USER_TOKEN"
}
```

You can override the default user and group field names. The default value for `UserNameAttributeField` is "user". The default value for `GroupAttributeField` is "groups".

Next, call `create-index` using the input file. For example, if the name of your JSON file is `create-index-openid.json`, you can use the following:

```
aws kendra create-index --cli-input-json file://create-index-openid.json
```

Python

```
response = kendra.create_index(
    Name='user-context',
    Edition='ENTERPRISE_EDITION',
    RoleArn='arn:aws:iam::account-id:role:/my-role',
    UserTokenConfigurations=[
        {
            "JwtTokenTypeConfiguration": {
                "KeyLocation": "URL",
                "Issuer": "optional: specify the issuer url",
                "ClaimRegex": "optional: regex to validate claims in the token",
                "UserNameAttributeField": "optional: user",
                "GroupAttributeField": "optional: group",
                "URL": "https://example.com/.well-known/jwks.json"
            }
        }
    ],
    UserContextPolicy='USER_TOKEN'
)
```

Using a JSON Web Token (JWT) with a shared secret

The following examples show how to use JSON Web Token (JWT) with a shared secret token for user access control when you create an index.

Console

1. Choose **Create index** to start creating a new index.
2. On the **Specify index details** page, give your index a name and a description.

3. For **IAM role**, select a role or select **Create a new role** to and specify a role name to create a new role. The IAM role will have the prefix "AmazonKendra-".
4. Leave all of the other fields at their defaults. Choose **Next**.
5. In the **Configure user access control** page, under **Access control settings**, choose **Yes** to use tokens for access control.
6. Under **Token configuration**, select **JWT with shared secret** as the **Token type**.
7. Under **Parameters for signing shared secret**, choose the **Type of secret**. You can use an existing AWS Secrets Manager shared secret or create a new shared secret.

   To create a new shared secret, choose **New** and then follow these steps:

   a. Under **New AWS Secrets Manager secret**, specify a **Secret name**. The prefix `AmazonKendra-` will be added when you save the public key.
   b. Specify a **Key ID**. The key id is a hint that indicates which key was used to secure the JSON web signature of the token.
   c. Choose the signing **Algorithm** for the token. This is the cryptographic algorithm used to secure the ID token. For more information on RSA, see [RSA Cryptography](https://tools.ietf.org/html/rfc3447).
   d. Specify a **Shared secret** by entering a base64 URL encoded secret. You can also select **Generate secret** to have a secret generated for you. You must ensure the secret is a base64 URL encoded secret.
   e. (*Optional*) Specify when the shared secret is valid. You can specify the date and time a secret is valid from, valid to, or both. The secret will be valid in the interval specified.
   f. Select **Save secret** to save the new secret.
8. (*Optional*) Under **Advanced configuration**:

   a. Specify a **Username** to use in the ACL check.
   b. Specify one or more **Groups** to use in the ACL check.
   c. Specify the **Issuer** that will validate the token issuer.
   d. Specify the **Claim ID(s)**. You must specify a regular expression that matches the audience in the JWT.
9. In the **Provisioning details** page, choose **Developer edition**.
10. Choose **Create** to create your index.
11. Wait for your index to be created. Amazon Kendra provisions the hardware for your index. This operation can take some time.

CLI

You can use JWT token with a shared secret inside of AWS Secrets Manager. The secret must be a base64 URL encoded secret. You need the Secrets Manager ARN, and your Amazon Kendra role must have access to `GetSecretValue` on the Secrets Manager resource. If you are encrypting the Secrets Manager resource with AWS KMS, the role must also have access to the decrypt action.

To create an index with the AWS CLI using a JSON input file, first create a JSON file with your desired parameters:

```
{
    "Name": "user-context",
    "Edition": "ENTERPRISE_EDITION",
    "RoleArn": "arn:aws:iam::account-id:role:/my-role",
    "UserTokenConfigurations": [
        {
            "JwtTokenTypeConfiguration": {
                "KeyLocation": "SECRET_MANAGER",
                "Issuer": "optional: specify the issuer url",
                "ClaimRegex": "optional: regex to validate claims in the token",
                "UserNameAttributeField": "optional: user",
                "GroupAttributeField": "optional: group",
                "SecretManagerArn": "arn:aws:secretsmanager:us-west-2:account
 id:secret:/my-user-context-secret
            }
        }
    ],
    "UserContextPolicy": "USER_TOKEN"
}
```

You can override the default user and group field names. The default value for `UserNameAttributeField` is "user". The default value for `GroupAttributeField` is "groups".

Next, call `create-index` using the input file. For example, if the name of your JSON file is `create-index-openid.json`, you can use the following:

```
aws kendra create-index --cli-input-json file://create-index-openid.json
```

The secret must have the following format in AWS Secrets Manager:

```
{
  "keys": [
    {
      "kid": "key_id",
      "alg": "HS256|HS384|HS512",
      "kty": "OCT",
      "use": "sig", //this value can be sig only for now
      "k": "secret",
      "nbf":"ISO1806 date format"
      "exp":"ISO1806 date format"
    }
  ]
}
```

For more information about JWT, see jwt.io.

Python

You can use JWT token with a shared secret inside of AWS Secrets Manager. The secret must be a base64 URL encoded secret. You need the Secrets Manager ARN, and your Amazon Kendra role must have access to `GetSecretValue` on the Secrets Manager resource. If you are encrypting the Secrets Manager resource with AWS KMS, the role must also have access to the decrypt action.

```
response = kendra.create_index(
    Name='user-context',
    Edition='ENTERPRISE_EDITION',
    RoleArn='arn:aws:iam::account-id:role:/my-role',
    UserTokenConfigurations=[
        {
            "JwtTokenTypeConfiguration": {
                "KeyLocation": "URL",
                "Issuer": "optional: specify the issuer url",
                "ClaimRegex": "optional: regex to validate claims in the token",
                "UserNameAttributeField": "optional: user",
                "GroupAttributeField": "optional: group",
                "SecretManagerArn": "arn:aws:secretsmanager:us-west-2:account
 id:secret:/my-user-context-secret"
            }
        }
    ],
    UserContextPolicy='USER_TOKEN'
```

```
)
```

Using a JSON Web Token (JWT) with a public key

The following examples show how to use JSON Web Token (JWT) with a public key for user access control when you create an index. For more information about JWT, see [jwt.io](jwt.io).

Console

1. Choose **Create index** to start creating a new index.
2. On the **Specify index details** page, give your index a name and a description.
3. For **IAM role**, select a role or select **Create a new role** to and specify a role name to create a new role. The IAM role will have the prefix "AmazonKendra-".
4. Leave all of the other fields at their defaults. Choose **Next**.
5. In the **Configure user access control** page, under **Access control settings**, choose **Yes** to use tokens for access control.
6. Under **Token configuration**, select **JWT with public key** as the **Token type**.
7. Under **Parameters for signing public key**, choose the **Type of secret**. You can use an existing AWS Secrets Manager secret or create a new secret.

   To create a new secret, choose **New** and then follow these steps:

   a. Under **New AWS Secrets Manager secret**, specify a **Secret name**. The prefix `AmazonKendra-` will be added when you save the public key.
   b. Specify a **Key ID**. The key id is a hint that indicates which key was used to secure the JSON web signature of the token.
   c. Choose the signing **Algorithm** for the token. This is the cryptographic algorithm used to secure the ID token. For more information on RSA, see RSA Cryptography.
   d. Under **Certificate attributes**, specify an *optional* **Certificate chain**. The certificate chain is made up of a list of certificates. It begins with a server's certificate and terminates with the root certificate.
   e. *Optional* Specify the **Thumbprint or fingerprint**. It should be is a hash of a certificate, computed over all certificate data and its signature.
   f. Specify the **Exponent**. This is the exponent value for the RSA public key. It is represented as a Base64urlUInt-encoded value.

   g. Specify the **Modulus**. This is the exponent value for the RSA public key. It is represented as a Base64urlUInt-encoded value.

   h. Select **Save key** to save the new key.

8. *Optional* Under **Advanced configuration**:

   a. Specify a **Username** to use in the ACL check.

   b. Specify one or more **Groups** to use in the ACL check.

   c. Specify the **Issuer** that will validate the token issuer.

   d. Specify the **Client Id(s)**. You must specify a regular expression that match the audience in the JWT.

9. In the **Provisioning details** page, choose **Developer edition**.

10. Choose **Create** to create your index.

11. Wait for your index to be created. Amazon Kendra provisions the hardware for your index. This operation can take some time.

CLI

You can use JWT with a public key inside of a AWS Secrets Manager. You need the Secrets Manager ARN, and your Amazon Kendra role must have access to `GetSecretValue` on the Secrets Manager resource. If you are encrypting the Secrets Manager resource with AWS KMS, the role must also have access to the decrypt action.

To create an index with the AWS CLI using a JSON input file, first create a JSON file with your desired parameters:

```
{
    "Name": "user-context",
    "Edition": "ENTERPRISE_EDITION",
    "RoleArn": "arn:aws:iam::account id:role:/my-role",
    "UserTokenConfigurationList": [
        {
            "JwtTokenTypeConfiguration": {
                "KeyLocation": "SECRET_MANAGER",
                "Issuer": "optional: specify the issuer url",
                "ClaimRegex": "optional: regex to validate claims in the token",
                "UserNameAttributeField": "optional: user",
                "GroupAttributeField": "optional: group",
```

```
                "SecretManagerArn": "arn:aws:secretsmanager:us-west-2:account
 id:secret:/my-user-context-secret
            }
        }
    ],    "UserContextPolicy": "USER_TOKEN"
}
```

You can override the default user and group field names. The default value for `UserNameAttributeField` is "user". The default value for `GroupAttributeField` is "groups".

Next, call `create-index` using the input file. For example, if the name of your JSON file is `create-index-openid.json`, you can use the following:

```
aws kendra create-index --cli-input-json file://create-index-openid.json
```

The secret must have the following format in Secrets Manager:

```
{
  "keys": [
    {
      "alg": "RS256|RS384|RS512",
      "kty": "RSA", //this can be RSA only for now
      "use": "sig", //this value can be sig only for now
      "n": "modulus of standard pem",
      "e": "exponent of standard pem",
      "kid": "key_id",
      "x5t": "certificate thumprint for x.509 cert",
      "x5c": [
        "certificate chain"
      ]
    }
  ]
}
```

For more information about JWT, see jwt.io.

Python

```
response = kendra.create_index(
    Name='user-context',
    Edition='ENTERPRISE_EDITION',
```

```
    RoleArn='arn:aws:iam::account id:role:/my-role',
    UserTokenConfigurationList=[
        {
            "JwtTokenTypeConfiguration": {
                "KeyLocation": "URL",
                "Issuer": "optional: specify the issuer url",
                "ClaimRegex": "optional: regex to validate claims in the token",
                "UserNameAttributeField": "optional: user",
                "GroupAttributeField": "optional: group",
                "SecretManagerArn": "arn:aws:secretsmanager:us-west-2:account
 id:secret:/my-user-context-secret"
            }
        }
    ],
    UserContextPolicy='USER_TOKEN'
)
```

Using JSON

The following examples show how to use JSON for user access control when you create an index.

> **Warning**
>
> The JSON token is a non-validated payload. This should only be used when requests to Amazon Kendra come from a trusted server and never from a browser.

Console

1. Choose **Create index** to start creating a new index.
2. On the **Specify index details** page, give your index a name and a description.
3. For **IAM role**, select a role or select **Create a new role** to and specify a role name to create a new role. The IAM role will have the prefix "AmazonKendra-".
4. Leave all of the other fields at their defaults. Choose **Next**.
5. In the **Configure user access control** page, under **Access control settings**, choose **Yes** to use tokens for access control.
6. Under **Token configuration**, select **JSON** as the **Token type**.
7. Specify a **User name** to use in the ACL check.

8. Specify one or more **Groups** to use in the ACL check.
9. Choose **Next**.
10. In the **Provisioning details** page, choose **Developer edition**.
11. Choose **Create** to create your index.
12. Wait for your index to be created. Amazon Kendra provisions the hardware for your index. This operation can take some time.

CLI

To create an index with the AWS CLI using a JSON input file, first create a JSON file with your desired parameters:

```
{
    "Name": "user-context",
    "Edition": "ENTERPRISE_EDITION",
    "RoleArn": "arn:aws:iam::account-id:role:/my-role",
    "UserTokenConfigurations": [
        {
            "JsonTokenTypeConfiguration": {
                "UserNameAttributeField": "user",
                "GroupAttributeField": "group"
            }
        }
    ],
    "UserContextPolicy": "USER_TOKEN"
}
```

Next, call `create-index` using the input file. For example, if the name of your JSON file is `create-index-openid.json`, you can use the following:

```
aws kendra create-index --cli-input-json file://create-index-openid.json
```

If you are not using Open ID for AWS IAM Identity Center, you can send us the token in JSON format. If you do, you must specify which field in the JSON token contains the user name and which field contains the groups. The group field values must be a JSON string array. For example, if you are using SAML, your token would be similar to the following:

```
{
```

```
      "username" : "user1",
      "groups": [
         "group1",
         "group2"
      ]
}
```

The `TokenConfiguration` would specify the user name and group field names:

```
{
    "UserNameAttributeField":"username",
    "GroupAttributeField":"groups"
}
```

Python

```
response = kendra.create_index(
    Name='user-context',
    Edition='ENTERPRISE_EDITION',
    RoleArn='arn:aws:iam::account-id:role:/my-role',
    UserTokenConfigurations=[
        {
            "JwtTokenTypeConfiguration": {
                "UserNameAttributeField": "user",
                "GroupAttributeField": "group",
            }
        }
    ],
    UserContextPolicy='USER_TOKEN'
)
```

Creating a data source connector

You can create a data source connector for Amazon Kendra to connect to and index your documents. Amazon Kendra can connect to Microsoft SharePoint, Google Drive, and many other providers. When you create a data source connector, you give Amazon Kendra the configuration information required to connect to your source repository. Unlike adding documents directly to an index, you can periodically scan the data source to update the index.

For example, say that you have a repository of tax documents stored in an Amazon S3 bucket. From time to time, existing documents are changed and new documents are added to the repository. If you add the repository to Amazon Kendra as a data source, you can keep your index up to date by setting up periodic synchronizations between your data source and index.

You can choose to update an index manually using the console or the StartDataSourceSyncJob API. Otherwise, you set up a schedule to update an index and have it synchronize with your data source.

An index can have more than one data source. Each data source can have its own update schedule. For example, you might update the index of your working documents daily, or even hourly, while updating your archived documents manually whenever the archive changes.

If you want to alter your document metadata or attributes and content during the document ingestion process, see Amazon Kendra Custom Document Enrichment.

> **Note**
>
> Each document ID must be unique per index. You cannot create a data source to index your documents with their unique IDs and then use the `BatchPutDocument` API to index the same documents, or vice versa. You can delete a data source and then use the `BatchPutDocument` API to index the same documents, or vice versa. Using the `BatchPutDocument` and `BatchDeleteDocument` APIs in combination with an Amazon Kendra data source connector for the same set of documents could cause inconsistencies with your data. Instead, we recommend using the Amazon Kendra custom data source connector.

> **Note**
>
> Files added to the index must be in a UTF-8 encoded byte stream. For more information on documents in Amazon Kendra, see Documents.

Setting an update schedule

Configure your data source to periodically update with the console or by using the `Schedule` parameter when you create or update a data source. The content of the parameter is a string that holds either a `cron`-format schedule string or an empty string to indicate that the index is updated on demand. For the format of a cron expression, see Schedule Expressions for Rules in the *Amazon CloudWatch Events User Guide*. Amazon Kendra supports only cron expressions. It doesn't support rate expressions.

Setting a language

You can index all your documents in a data source in a supported language. You specify the language code for all your documents in your data source when you call CreateDataSource. If a document doesn't have a language code specified in a metadata field, the document is indexed using the language code that's specified for all documents at the data source level. If you don't specify a language, Amazon Kendra indexes documents in a data source in English by default. For more information on supported languages, including their codes, see Adding documents in languages other than English.

You index all your documents in a data source in a supported language using the console. Go to **Data sources** and edit your data source or **Add data source** if you're adding a new data source. On the **Specify data source details** page, choose a language from the dropdown **Language**. You select **Update** or continue to enter the configuration information to connect to your data source.

Data source connectors

This section shows you how to connect Amazon Kendra to supported databases and data source repositories using Amazon Kendra in the AWS Management Console and the Amazon Kendra APIs.

Topics

- Data source template schemas

- Adobe Experience Manager
- Alfresco
- Aurora (MySQL)
- Aurora (PostgreSQL)
- Amazon FSx (Windows)
- Amazon FSx (NetApp ONTAP)
- Amazon RDS/Aurora
- Amazon RDS (Microsoft SQL Server)
- Amazon RDS (MySQL)
- Amazon RDS (Oracle)
- Amazon RDS (PostgreSQL)
- Amazon S3
- Amazon Kendra Web Crawler
- Box
- Confluence
- Custom data source connector
- Dropbox
- Drupal
- GitHub
- Gmail
- Google Drive
- IBM DB2
- Jira
- Microsoft Exchange
- Microsoft OneDrive
- Microsoft SharePoint
- Microsoft SQL Server
- Microsoft Teams
- Microsoft Yammer
- MySQL

- Oracle Database
- PostgreSQL
- Quip
- Salesforce
- ServiceNow
- Slack
- Zendesk

Data source template schemas

The following are template schemas for data sources where templates are supported.

Topics

- Microsoft Exchange template schema
- Microsoft OneDrive template schema
- Microsoft SharePoint template schema
- Microsoft SQL Server template schema
- Microsoft Teams template schema
- Microsoft Yammer template schema
- MySQL template schema
- Oracle Database template schema
- PostgreSQL template schema
- Salesforce template schema
- ServiceNow template schema
- Slack template schema
- Zendesk template schema

Adobe Experience Manager template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the Adobe Experience Manager host URL, the authentication type, and whether you use Adobe Experience Manager (AEM) as a Cloud Service or AEM On-Premise as part of the connection configuration or repository endpoint details. Also, specify the type of data source as `AEM`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. For more information, see Adobe Experience Manager JSON schema.

The following table describes the parameters of the AEM JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.

Configuration	Description
aemUrl	The Adobe Experience Manager host URL. For example, if you use AEM On-Premise, you include the hostname and port: *https://hostname:port*. Or, if you use AEM as a Cloud Service, you can use the author URL: *https://author-xxxxxx-xxxxxxx.adobeaemcloud.com*.
authType	The type of authentication you use, whether `Basic` or `OAuth2`.
deploymentType	The type of Adobe Experience Manager that you use, either `CLOUD` or `ON_PREMISE` .
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
• page • asset	A list of objects that map the attributes or field names of your Adobe Experience Manager pages and assets to Amazon Kendra index field names. For more information, see [Mapping data source fields]().
additionalProperties	Additional configuration options for your content in your data source.

Configuration	Description
timeZoneId	If you use AEM On-Premise and the time zone of your server is different than the time zone of the Amazon Kendra AEM connector or index, you can specify the server time zone to align with the AEM connector or index. The default time zone for AEM On-Premise is the time zone of the Amazon Kendra AEM connector or index. The default time zone for AEM as a Cloud Service is Greenwich Mean Time.
• pageRootPaths • assetRootPaths	A list of root paths for pages and assets. For example, the root path for a page could be */content/sub* and the root path for an asset could be */content/sub/asset1*.
crawlAssets	`true` to crawl assets.
crawlPages	`true` to crawl pages.
• pagePathInclusionPatterns • pageNameInclusionPatterns • assetPathInclusionPatterns • assetTypeInclusionPatterns • assetNameInclusionPatterns	A list of regular expression patterns to include certain pages and assets in your Adobe Experience Manager data source. Pages and assets that match the patterns are included in the index. Pages and assets that don't match the patterns are excluded from the index. If a page or asset matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the content isn't included in the index.

Configuration	Description
• pagePathExclusionPatterns • pageNameExclusionPatterns • assetPathExclusionPatterns • assetTypeInclusionPatterns • assetNameInclusionPatterns	A list of regular expression patterns to exclude certain pages and assets in your Adobe Experience Manager data source. Pages and assets that match the patterns are excluded from the index. Pages and assets that don't match the patterns are included in the index. If a page or asset matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the content isn't included in the index.
pageComponents	A list of names for the specific page components that you want to index.
contentFragmentVariations	A list of names for the specific saved variation s of Adobe Experience Manager Content Fragments that you want to index.
type	The type of data source. Specify `AEM` as your data source type.

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
secretArn	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Adobe Experience Manager. For information on these key-value pairs, see Connection instructions for Adobe Experience Manager.
version	The version of this template that is currently supported.

Adobe Experience Manager JSON schema

```
{
```

```
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties":
  {
    "connectionConfiguration": {
      "type": "object",
      "properties":
      {
        "repositoryEndpointMetadata":
        {
          "type": "object",
          "properties":
          {
            "aemUrl":
            {
              "type": "string",
              "pattern": "https:.*"
            },
            "authType": {
              "type": "string",
              "enum": ["Basic", "OAuth2"]
            },
            "deploymentType": {
              "type": "string",
              "enum": ["CLOUD","ON_PREMISE"]
            }
          },
          "required":
          [
            "aemUrl",
            "authType",
            "deploymentType"
          ]
        }
      },
      "required":
      [
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties":
      {
```

```
        "page":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
                {
                  "type": "object",
                  "properties":
                  {
                    "indexFieldName":
                    {
                      "type": "string"
                    },
                    "indexFieldType":
                    {
                      "type": "string",
                      "enum":
                      [
                        "STRING",
                        "STRING_LIST",
                        "DATE",
                        "LONG"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
                    {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required":
                  [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
```

```
                  ]
                }
              ]
            }
          },
          "required":
          [
            "fieldMappings"
          ]
        },
        "asset":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
                {
                  "type": "object",
                  "properties":
                  {
                    "indexFieldName":
                    {
                      "type": "string"
                    },
                    "indexFieldType":
                    {
                      "type": "string",
                      "enum":
                      [
                        "STRING",
                        "STRING_LIST",
                        "DATE",
                        "LONG"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
```

```
                {
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                }
              },
              "required":
              [
                "indexFieldName",
                "indexFieldType",
                "dataSourceFieldName"
              ]
            }
          ]
        }
      },
      "required":
      [
        "fieldMappings"
      ]
    }
  }
},
"additionalProperties": {
  "type": "object",
  "properties":
  {
    "timeZoneId": {
      "type": "string",
      "enum": [
        "Africa/Abidjan",
        "Africa/Accra",
        "Africa/Addis_Ababa",
        "Africa/Algiers",
        "Africa/Asmara",
        "Africa/Asmera",
        "Africa/Bamako",
        "Africa/Bangui",
        "Africa/Banjul",
        "Africa/Bissau",
        "Africa/Blantyre",
        "Africa/Brazzaville",
        "Africa/Bujumbura",
        "Africa/Cairo",
        "Africa/Casablanca",
```

```
                "Africa/Ceuta",
                "Africa/Conakry",
                "Africa/Dakar",
                "Africa/Dar_es_Salaam",
                "Africa/Djibouti",
                "Africa/Douala",
                "Africa/El_Aaiun",
                "Africa/Freetown",
                "Africa/Gaborone",
                "Africa/Harare",
                "Africa/Johannesburg",
                "Africa/Juba",
                "Africa/Kampala",
                "Africa/Khartoum",
                "Africa/Kigali",
                "Africa/Kinshasa",
                "Africa/Lagos",
                "Africa/Libreville",
                "Africa/Lome",
                "Africa/Luanda",
                "Africa/Lubumbashi",
                "Africa/Lusaka",
                "Africa/Malabo",
                "Africa/Maputo",
                "Africa/Maseru",
                "Africa/Mbabane",
                "Africa/Mogadishu",
                "Africa/Monrovia",
                "Africa/Nairobi",
                "Africa/Ndjamena",
                "Africa/Niamey",
                "Africa/Nouakchott",
                "Africa/Ouagadougou",
                "Africa/Porto-Novo",
                "Africa/Sao_Tome",
                "Africa/Timbuktu",
                "Africa/Tripoli",
                "Africa/Tunis",
                "Africa/Windhoek",
                "America/Adak",
                "America/Anchorage",
                "America/Anguilla",
                "America/Antigua",
                "America/Araguaina",
```

```
                "America/Argentina/Buenos_Aires",
                "America/Argentina/Catamarca",
                "America/Argentina/ComodRivadavia",
                "America/Argentina/Cordoba",
                "America/Argentina/Jujuy",
                "America/Argentina/La_Rioja",
                "America/Argentina/Mendoza",
                "America/Argentina/Rio_Gallegos",
                "America/Argentina/Salta",
                "America/Argentina/San_Juan",
                "America/Argentina/San_Luis",
                "America/Argentina/Tucuman",
                "America/Argentina/Ushuaia",
                "America/Aruba",
                "America/Asuncion",
                "America/Atikokan",
                "America/Atka",
                "America/Bahia",
                "America/Bahia_Banderas",
                "America/Barbados",
                "America/Belem",
                "America/Belize",
                "America/Blanc-Sablon",
                "America/Boa_Vista",
                "America/Bogota",
                "America/Boise",
                "America/Buenos_Aires",
                "America/Cambridge_Bay",
                "America/Campo_Grande",
                "America/Cancun",
                "America/Caracas",
                "America/Catamarca",
                "America/Cayenne",
                "America/Cayman",
                "America/Chicago",
                "America/Chihuahua",
                "America/Ciudad_Juarez",
                "America/Coral_Harbour",
                "America/Cordoba",
                "America/Costa_Rica",
                "America/Creston",
                "America/Cuiaba",
                "America/Curacao",
                "America/Danmarkshavn",
```

```
                "America/Dawson",
                "America/Dawson_Creek",
                "America/Denver",
                "America/Detroit",
                "America/Dominica",
                "America/Edmonton",
                "America/Eirunepe",
                "America/El_Salvador",
                "America/Ensenada",
                "America/Fort_Nelson",
                "America/Fort_Wayne",
                "America/Fortaleza",
                "America/Glace_Bay",
                "America/Godthab",
                "America/Goose_Bay",
                "America/Grand_Turk",
                "America/Grenada",
                "America/Guadeloupe",
                "America/Guatemala",
                "America/Guayaquil",
                "America/Guyana",
                "America/Halifax",
                "America/Havana",
                "America/Hermosillo",
                "America/Indiana/Indianapolis",
                "America/Indiana/Knox",
                "America/Indiana/Marengo",
                "America/Indiana/Petersburg",
                "America/Indiana/Tell_City",
                "America/Indiana/Vevay",
                "America/Indiana/Vincennes",
                "America/Indiana/Winamac",
                "America/Indianapolis",
                "America/Inuvik",
                "America/Iqaluit",
                "America/Jamaica",
                "America/Jujuy",
                "America/Juneau",
                "America/Kentucky/Louisville",
                "America/Kentucky/Monticello",
                "America/Knox_IN",
                "America/Kralendijk",
                "America/La_Paz",
                "America/Lima",
```

```
                "America/Los_Angeles",
                "America/Louisville",
                "America/Lower_Princes",
                "America/Maceio",
                "America/Managua",
                "America/Manaus",
                "America/Marigot",
                "America/Martinique",
                "America/Matamoros",
                "America/Mazatlan",
                "America/Mendoza",
                "America/Menominee",
                "America/Merida",
                "America/Metlakatla",
                "America/Mexico_City",
                "America/Miquelon",
                "America/Moncton",
                "America/Monterrey",
                "America/Montevideo",
                "America/Montreal",
                "America/Montserrat",
                "America/Nassau",
                "America/New_York",
                "America/Nipigon",
                "America/Nome",
                "America/Noronha",
                "America/North_Dakota/Beulah",
                "America/North_Dakota/Center",
                "America/North_Dakota/New_Salem",
                "America/Nuuk",
                "America/Ojinaga",
                "America/Panama",
                "America/Pangnirtung",
                "America/Paramaribo",
                "America/Phoenix",
                "America/Port-au-Prince",
                "America/Port_of_Spain",
                "America/Porto_Acre",
                "America/Porto_Velho",
                "America/Puerto_Rico",
                "America/Punta_Arenas",
                "America/Rainy_River",
                "America/Rankin_Inlet",
                "America/Recife",
```

```
"America/Regina",
"America/Resolute",
"America/Rio_Branco",
"America/Rosario",
"America/Santa_Isabel",
"America/Santarem",
"America/Santiago",
"America/Santo_Domingo",
"America/Sao_Paulo",
"America/Scoresbysund",
"America/Shiprock",
"America/Sitka",
"America/St_Barthelemy",
"America/St_Johns",
"America/St_Kitts",
"America/St_Lucia",
"America/St_Thomas",
"America/St_Vincent",
"America/Swift_Current",
"America/Tegucigalpa",
"America/Thule",
"America/Thunder_Bay",
"America/Tijuana",
"America/Toronto",
"America/Tortola",
"America/Vancouver",
"America/Virgin",
"America/Whitehorse",
"America/Winnipeg",
"America/Yakutat",
"America/Yellowknife",
"Antarctica/Casey",
"Antarctica/Davis",
"Antarctica/DumontDUrville",
"Antarctica/Macquarie",
"Antarctica/Mawson",
"Antarctica/McMurdo",
"Antarctica/Palmer",
"Antarctica/Rothera",
"Antarctica/South_Pole",
"Antarctica/Syowa",
"Antarctica/Troll",
"Antarctica/Vostok",
"Arctic/Longyearbyen",
```

```
                "Asia/Aden",
                "Asia/Almaty",
                "Asia/Amman",
                "Asia/Anadyr",
                "Asia/Aqtau",
                "Asia/Aqtobe",
                "Asia/Ashgabat",
                "Asia/Ashkhabad",
                "Asia/Atyrau",
                "Asia/Baghdad",
                "Asia/Bahrain",
                "Asia/Baku",
                "Asia/Bangkok",
                "Asia/Barnaul",
                "Asia/Beirut",
                "Asia/Bishkek",
                "Asia/Brunei",
                "Asia/Calcutta",
                "Asia/Chita",
                "Asia/Choibalsan",
                "Asia/Chongqing",
                "Asia/Chungking",
                "Asia/Colombo",
                "Asia/Dacca",
                "Asia/Damascus",
                "Asia/Dhaka",
                "Asia/Dili",
                "Asia/Dubai",
                "Asia/Dushanbe",
                "Asia/Famagusta",
                "Asia/Gaza",
                "Asia/Harbin",
                "Asia/Hebron",
                "Asia/Ho_Chi_Minh",
                "Asia/Hong_Kong",
                "Asia/Hovd",
                "Asia/Irkutsk",
                "Asia/Istanbul",
                "Asia/Jakarta",
                "Asia/Jayapura",
                "Asia/Jerusalem",
                "Asia/Kabul",
                "Asia/Kamchatka",
                "Asia/Karachi",
```

```
                "Asia/Kashgar",
                "Asia/Kathmandu",
                "Asia/Katmandu",
                "Asia/Khandyga",
                "Asia/Kolkata",
                "Asia/Krasnoyarsk",
                "Asia/Kuala_Lumpur",
                "Asia/Kuching",
                "Asia/Kuwait",
                "Asia/Macao",
                "Asia/Macau",
                "Asia/Magadan",
                "Asia/Makassar",
                "Asia/Manila",
                "Asia/Muscat",
                "Asia/Nicosia",
                "Asia/Novokuznetsk",
                "Asia/Novosibirsk",
                "Asia/Omsk",
                "Asia/Oral",
                "Asia/Phnom_Penh",
                "Asia/Pontianak",
                "Asia/Pyongyang",
                "Asia/Qatar",
                "Asia/Qostanay",
                "Asia/Qyzylorda",
                "Asia/Rangoon",
                "Asia/Riyadh",
                "Asia/Saigon",
                "Asia/Sakhalin",
                "Asia/Samarkand",
                "Asia/Seoul",
                "Asia/Shanghai",
                "Asia/Singapore",
                "Asia/Srednekolymsk",
                "Asia/Taipei",
                "Asia/Tashkent",
                "Asia/Tbilisi",
                "Asia/Tehran",
                "Asia/Tel_Aviv",
                "Asia/Thimbu",
                "Asia/Thimphu",
                "Asia/Tokyo",
                "Asia/Tomsk",
```

```
                "Asia/Ujung_Pandang",
                "Asia/Ulaanbaatar",
                "Asia/Ulan_Bator",
                "Asia/Urumqi",
                "Asia/Ust-Nera",
                "Asia/Vientiane",
                "Asia/Vladivostok",
                "Asia/Yakutsk",
                "Asia/Yangon",
                "Asia/Yekaterinburg",
                "Asia/Yerevan",
                "Atlantic/Azores",
                "Atlantic/Bermuda",
                "Atlantic/Canary",
                "Atlantic/Cape_Verde",
                "Atlantic/Faeroe",
                "Atlantic/Faroe",
                "Atlantic/Jan_Mayen",
                "Atlantic/Madeira",
                "Atlantic/Reykjavik",
                "Atlantic/South_Georgia",
                "Atlantic/St_Helena",
                "Atlantic/Stanley",
                "Australia/ACT",
                "Australia/Adelaide",
                "Australia/Brisbane",
                "Australia/Broken_Hill",
                "Australia/Canberra",
                "Australia/Currie",
                "Australia/Darwin",
                "Australia/Eucla",
                "Australia/Hobart",
                "Australia/LHI",
                "Australia/Lindeman",
                "Australia/Lord_Howe",
                "Australia/Melbourne",
                "Australia/NSW",
                "Australia/North",
                "Australia/Perth",
                "Australia/Queensland",
                "Australia/South",
                "Australia/Sydney",
                "Australia/Tasmania",
                "Australia/Victoria",
```

```
                "Australia/West",
                "Australia/Yancowinna",
                "Brazil/Acre",
                "Brazil/DeNoronha",
                "Brazil/East",
                "Brazil/West",
                "CET",
                "CST6CDT",
                "Canada/Atlantic",
                "Canada/Central",
                "Canada/Eastern",
                "Canada/Mountain",
                "Canada/Newfoundland",
                "Canada/Pacific",
                "Canada/Saskatchewan",
                "Canada/Yukon",
                "Chile/Continental",
                "Chile/EasterIsland",
                "Cuba",
                "EET",
                "EST5EDT",
                "Egypt",
                "Eire",
                "Etc/GMT",
                "Etc/GMT+0",
                "Etc/GMT+1",
                "Etc/GMT+10",
                "Etc/GMT+11",
                "Etc/GMT+12",
                "Etc/GMT+2",
                "Etc/GMT+3",
                "Etc/GMT+4",
                "Etc/GMT+5",
                "Etc/GMT+6",
                "Etc/GMT+7",
                "Etc/GMT+8",
                "Etc/GMT+9",
                "Etc/GMT-0",
                "Etc/GMT-1",
                "Etc/GMT-10",
                "Etc/GMT-11",
                "Etc/GMT-12",
                "Etc/GMT-13",
                "Etc/GMT-14",
```

```
                "Etc/GMT-2",
                "Etc/GMT-3",
                "Etc/GMT-4",
                "Etc/GMT-5",
                "Etc/GMT-6",
                "Etc/GMT-7",
                "Etc/GMT-8",
                "Etc/GMT-9",
                "Etc/GMT0",
                "Etc/Greenwich",
                "Etc/UCT",
                "Etc/UTC",
                "Etc/Universal",
                "Etc/Zulu",
                "Europe/Amsterdam",
                "Europe/Andorra",
                "Europe/Astrakhan",
                "Europe/Athens",
                "Europe/Belfast",
                "Europe/Belgrade",
                "Europe/Berlin",
                "Europe/Bratislava",
                "Europe/Brussels",
                "Europe/Bucharest",
                "Europe/Budapest",
                "Europe/Busingen",
                "Europe/Chisinau",
                "Europe/Copenhagen",
                "Europe/Dublin",
                "Europe/Gibraltar",
                "Europe/Guernsey",
                "Europe/Helsinki",
                "Europe/Isle_of_Man",
                "Europe/Istanbul",
                "Europe/Jersey",
                "Europe/Kaliningrad",
                "Europe/Kiev",
                "Europe/Kirov",
                "Europe/Kyiv",
                "Europe/Lisbon",
                "Europe/Ljubljana",
                "Europe/London",
                "Europe/Luxembourg",
                "Europe/Madrid",
```

```
                "Europe/Malta",
                "Europe/Mariehamn",
                "Europe/Minsk",
                "Europe/Monaco",
                "Europe/Moscow",
                "Europe/Nicosia",
                "Europe/Oslo",
                "Europe/Paris",
                "Europe/Podgorica",
                "Europe/Prague",
                "Europe/Riga",
                "Europe/Rome",
                "Europe/Samara",
                "Europe/San_Marino",
                "Europe/Sarajevo",
                "Europe/Saratov",
                "Europe/Simferopol",
                "Europe/Skopje",
                "Europe/Sofia",
                "Europe/Stockholm",
                "Europe/Tallinn",
                "Europe/Tirane",
                "Europe/Tiraspol",
                "Europe/Ulyanovsk",
                "Europe/Uzhgorod",
                "Europe/Vaduz",
                "Europe/Vatican",
                "Europe/Vienna",
                "Europe/Vilnius",
                "Europe/Volgograd",
                "Europe/Warsaw",
                "Europe/Zagreb",
                "Europe/Zaporozhye",
                "Europe/Zurich",
                "GB",
                "GB-Eire",
                "GMT",
                "GMT0",
                "Greenwich",
                "Hongkong",
                "Iceland",
                "Indian/Antananarivo",
                "Indian/Chagos",
                "Indian/Christmas",
```

```
"Indian/Cocos",
"Indian/Comoro",
"Indian/Kerguelen",
"Indian/Mahe",
"Indian/Maldives",
"Indian/Mauritius",
"Indian/Mayotte",
"Indian/Reunion",
"Iran",
"Israel",
"Jamaica",
"Japan",
"Kwajalein",
"Libya",
"MET",
"MST7MDT",
"Mexico/BajaNorte",
"Mexico/BajaSur",
"Mexico/General",
"NZ",
"NZ-CHAT",
"Navajo",
"PRC",
"PST8PDT",
"Pacific/Apia",
"Pacific/Auckland",
"Pacific/Bougainville",
"Pacific/Chatham",
"Pacific/Chuuk",
"Pacific/Easter",
"Pacific/Efate",
"Pacific/Enderbury",
"Pacific/Fakaofo",
"Pacific/Fiji",
"Pacific/Funafuti",
"Pacific/Galapagos",
"Pacific/Gambier",
"Pacific/Guadalcanal",
"Pacific/Guam",
"Pacific/Honolulu",
"Pacific/Johnston",
"Pacific/Kanton",
"Pacific/Kiritimati",
"Pacific/Kosrae",
```

```
                "Pacific/Kwajalein",
                "Pacific/Majuro",
                "Pacific/Marquesas",
                "Pacific/Midway",
                "Pacific/Nauru",
                "Pacific/Niue",
                "Pacific/Norfolk",
                "Pacific/Noumea",
                "Pacific/Pago_Pago",
                "Pacific/Palau",
                "Pacific/Pitcairn",
                "Pacific/Pohnpei",
                "Pacific/Ponape",
                "Pacific/Port_Moresby",
                "Pacific/Rarotonga",
                "Pacific/Saipan",
                "Pacific/Samoa",
                "Pacific/Tahiti",
                "Pacific/Tarawa",
                "Pacific/Tongatapu",
                "Pacific/Truk",
                "Pacific/Wake",
                "Pacific/Wallis",
                "Pacific/Yap",
                "Poland",
                "Portugal",
                "ROK",
                "Singapore",
                "SystemV/AST4",
                "SystemV/AST4ADT",
                "SystemV/CST6",
                "SystemV/CST6CDT",
                "SystemV/EST5",
                "SystemV/EST5EDT",
                "SystemV/HST10",
                "SystemV/MST7",
                "SystemV/MST7MDT",
                "SystemV/PST8",
                "SystemV/PST8PDT",
                "SystemV/YST9",
                "SystemV/YST9YDT",
                "Turkey",
                "UCT",
                "US/Alaska",
```

```
                "US/Aleutian",
                "US/Arizona",
                "US/Central",
                "US/East-Indiana",
                "US/Eastern",
                "US/Hawaii",
                "US/Indiana-Starke",
                "US/Michigan",
                "US/Mountain",
                "US/Pacific",
                "US/Samoa",
                "UTC",
                "Universal",
                "W-SU",
                "WET",
                "Zulu",
                "EST",
                "HST",
                "MST",
                "ACT",
                "AET",
                "AGT",
                "ART",
                "AST",
                "BET",
                "BST",
                "CAT",
                "CNT",
                "CST",
                "CTT",
                "EAT",
                "ECT",
                "IET",
                "IST",
                "JST",
                "MIT",
                "NET",
                "NST",
                "PLT",
                "PNT",
                "PRT",
                "PST",
                "SST",
                "VST"
```

```
          ]
        },
        "pageRootPaths":
        {
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "assetRootPaths":
        {
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "crawlAssets":
        {
          "type": "boolean"
        },
        "crawlPages":
        {
          "type": "boolean"
        },
        "pagePathInclusionPatterns":
        {
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "pagePathExclusionPatterns":
        {
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "pageNameInclusionPatterns":
        {
```

```
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "pageNameExclusionPatterns":
        {
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "assetPathInclusionPatterns":
        {
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "assetPathExclusionPatterns":
        {
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "assetTypeInclusionPatterns":
        {
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "assetTypeExclusionPatterns":
        {
          "type": "array",
          "items":
          {
            "type": "string"
```

```
          }
        },
        "assetNameInclusionPatterns":
        {
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "assetNameExclusionPatterns":
        {
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "pageComponents": {
          "type": "array",
          "items": {
            "type": "object"
            }
        },
        "contentFragmentVariations": {
          "type": "array",
          "items": {
            "type": "object"
          }
        },
        "cugExemptedPrincipals": {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      },
      "required":
      []
    },
    "type": {
      "type": "string",
      "pattern": "AEM"
    },
```

```
    "enableIdentityCrawler": {
      "type": "boolean"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL",
        "CHANGE_LOG"
      ]
    },
    "secretArn": {
      "type": "string",
      "minLength": 20,
      "maxLength": 2048
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
    "connectionConfiguration",
    "repositoryConfigurations",
    "syncMode",
    "additionalProperties",
    "secretArn",
    "type"
  ]
}
```

Amazon FSx (Windows) template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the file system ID as part of the connection configuration or repository endpoint details. You must also specify the type of data source as FSX, a secret for your authentication credentials, and other necessary configurations. You then specify TEMPLATE as the Type when you call CreateDataSource.

You can use the template provided in this developer guide. See Amazon FSx (Windows) JSON schema.

The following table describes the parameters of the Amazon FSx (Windows) JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
fileSystemId	The identifier of the Amazon FSx file system. You can find your file system ID on the File Systems dashboard in the Amazon FSx console.
fileSystemType	The Amazon FSx file system type. To use Windows File Server as your type of file system, specify `WINDOWS`.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
All	A list of objects that map attributes or field names of your files in your Amazon FSx data source to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source.
isCrawlAcl	`true` to crawl the access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is

Configuration	Description
	used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
inclusionPatterns	A list of regular expression patterns to *include* certain files in your Amazon FSx data source. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.
exclusionPatterns	A list of regular expression patterns to *exclude* certain files in your Amazon FSx data source. Files that match the patterns are excluded from the index. Files that don't match the patterns are included in the index. If a file matches both an exclusion and inclusion pattern, the exclusion pattern takes precedenc e and the file isn't included in the index.
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
type	The type of data source. For Windows file system data sources, specify `FSX`.

Amazon FSx (Windows) JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "fileSystemId": {
              "type": "string",
              "pattern": "fs-.*"
            },
            "fileSystemType": {
              "type": "string",
              "pattern": "WINDOWS"
```

```
            }
          },
          "required": ["fileSystemId", "fileSystemType"]
        }
      }
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "All": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string",
                      "enum": ["STRING", "STRING_LIST", "DATE"]
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": ["fieldMappings"]
        }
```

```
      },
      "required": ["All"]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "isCrawlAcl": {
          "type": "boolean"
        },
        "exclusionPatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionPatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      },
      "required": []
    },
    "enableIdentityCrawler": {
      "type": "boolean"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL"
      ]
    },
    "type" : {
      "type" : "string",
      "pattern": "FSX"
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
```

```
      }
    ]
  },
  "required": [
    "connectionConfiguration",
    "repositoryConfigurations",
    "syncMode",
    "enableIdentityCrawler",
    "additionalProperties",
    "type"
  ]
}
```

Amazon FSx (NetApp ONTAP) template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the file system ID and the storage virtual machine (SVM) as part of the connection configuration or repository endpoint details. You must also specify the type of data source as `FSXONTAP`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Amazon FSx (NetApp ONTAP) JSON schema.

The following table describes the parameters of the Amazon FSx (NetApp ONTAP) JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
fileSystemId	The identifier of the Amazon FSx file system. You can find your file system ID on the File Systems dashboard in the Amazon FSx console. For information about how to create a file system in the Amazon FSx console for NetApp ONTAP, see Getting Started Guide

Configuration	Description
	for NetApp ONTAP in the *FSx for ONTAP User Guide*.
fileSystemType	The Amazon FSx file system type. To use NetApp ONTAP as your type of file system, specify `ONTAP`.
svmId	The identifier of storage virtual machine (SVM) used with your Amazon FSx file system for NetApp ONTAP. You can find your SVM ID by going to the File Systems dashboard in the Amazon FSx console, selecting your file system ID, and then selecting **Storage virtual machines**. For information about how to create a file system in the Amazon FSx console for NetApp ONTAP, see Getting Started Guide for NetApp ONTAP in the *FSx for ONTAP User Guide*.
protocolType	Whether you use the Common Internet File System (CIFS) protocol for Windows, or the Network File System (NFS) protocol for Linux.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
file	A list of objects that map attributes or field names of your files in your Amazon FSx data source to Amazon Kendra index field names. For more information, see Mapping data source fields. The data source field names must exist in your files custom metadata.
additionalProperties	Additional configuration options for your content in your data source.

Configuration	Description
crawlAcl	`true` to crawl the access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see [User context filtering]().
inclusionPatterns	A list of regular expression patterns to *include* certain files in your Amazon FSx data source. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.
exclusionPatterns	A list of regular expression patterns to *exclude* certain files in your Amazon FSx data source. Files that match the patterns are excluded from the index. Files that don't match the patterns are included in the index. If a file matches both an exclusion and inclusion pattern, the exclusion pattern takes precedenc e and the file isn't included in the index.
type	The type of data source. For NetApp ONTAP file system data sources, specify `FSXONTAP`.

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

<table>
<tr><th>Configuration</th><th>Description</th></tr>
<tr><td>secretArn</td><td>The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Amazon FSx file system. The secret must contain a JSON structure with the following keys:
<pre>{
    "username": " user@corp.example.
com ",
    "password": " password"
}</pre>
If you use the NFS protocol for your Amazon FSx file system, the secret is stored in a JSON structure with the following keys:
<pre>{
    "leftId": "left ID",
    "rightId": " right ID",
    "preSharedKey": " pre-shared key "
}</pre></td></tr>
</table>

Amazon FSx (NetApp ONTAP) JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "fileSystemId": {
              "type": "string",
```

```
                "pattern": "^(fs-[0-9a-f]{8,21})$"
            },
            "fileSystemType": {
              "type": "string",
              "enum": ["ONTAP"]
            },
            "svmId": {
              "type": "string",
              "pattern": "^(svm-[0-9a-f]{17,21})$"
            },
            "protocolType": {
              "type": "string",
              "enum": [
                "CIFS",
                "NFS"
              ]
            }
          },
          "required": [
            "fileSystemId",
            "fileSystemType"
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "file": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string",
                      "pattern": "^([a-zA-Z_]{1,20})$"
                    },
```

```
                    "indexFieldType": {
                      "type": "string",
                      "enum": [
                        "STRING",
                        "STRING_LIST",
                        "DATE",
                        "LONG"
                      ]
                    },
                    "dataSourceFieldName": {
                      "type": "string",
                      "pattern": "^([a-zA-Z_]{1,20})$"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ],
              "maxItems": 50
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
        "file"
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "crawlAcl": {
          "type": "boolean"
        },
        "inclusionPatterns": {
```

```
          "type": "array",
          "items": {
            "type": "string",
            "maxLength": 30
          },
          "maxItems": 100
        },
        "exclusionPatterns": {
          "type": "array",
          "items": {
            "type": "string",
            "maxLength": 30
          },
          "maxItems": 100
        }
      }
    },
    "type": {
      "type": "string",
      "pattern": "FSXONTAP"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL"
      ]
    },
    "secretArn": {
      "type": "string",
      "pattern": "arn:aws:secretsmanager:.*"
    }
  },
  "required": [
    "connectionConfiguration",
    "repositoryConfigurations",
    "additionalProperties",
    "secretArn",
    "type"
  ]
}
```

Alfresco template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the Alfresco site ID, repository URL, user interface URL, authentication type, whether you use cloud or on-premises, and the type of content you want to crawl. You provide this as a part of the connection configuration or repository endpoint details. Also specify the type of data source as `ALFRESCO`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Alfresco JSON schema.

The following table describes the parameters of the Alfresco JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
siteId	The identifier of the Alfresco site.
repoUrl	The URL of your Alfresco repository. You can get the repository URL from your Alfresco administrator. For example, if you use Alfresco Cloud (PaaS), the repository URL could be *https://company.alfrescocloud.com*. Or, if you use Alfresco On-Premises, the repository URL could be *https://company-alfresco-instance.company-domain.suffix:port*.
webAppUrl	The URL of your Alfresco user interface. You can get the Alfresco user interface URL from your Alfresco administrator. For example, the user interface URL could be *https://example.com*.
repositoryAdditionalProperties	Additional properties to connect with the repository/data source endpoint.

Configuration	Description
authType	The type of authentication that you use, whether `OAuth2` or `Basic`.
type (deployment)	The type of Alfresco that you use, whether `PAAS` or `ON-PREM`.
crawlType	The type of content that you want to crawl, whether `ASPECT` (content marked with 'Aspects' in Alfresco), `SITE_ID` (content within a specific Alfresco site), or `ALL_SITES` (content across all your Alfresco sites).
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
• document • comment	A list of objects that map the attributes or field names of your Alfresco documents and comments to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source.
aspectName	The name of a specific 'Aspect' that you want to index.
aspectProperties	A list of specific 'Aspect' content properties that you want to index.
enableFineGrainedControl	`true` to crawl 'Aspects'.
isCrawlComment	`true` to crawl comments.

Configuration	Description
• inclusionFileNamePatterns • inclusionFileTypePatterns • inclusionFilePathPatterns	A list of regular expression patterns to include certain files in your Alfresco data source. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the file isn't included in the index.
• exclusionFileNamePatterns • exclusionFileTypePatterns • exclusionFilePathPatterns	A list of regular expression patterns to exclude certain files in your Alfresco data source. Files that match the patterns are excluded from the index. Files that don't match the patterns are included in the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the file isn't included in the index.
type	The type of data source. Specify `ALFRESCO` as your data source type.

<table>
<tr><th>Configuration</th><th>Description</th></tr>
<tr><td>secretArn</td><td>The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs that are required to connect to your Alfresco. The secret must contain a JSON structure with the following keys:

If using basic authentication:
<pre>{
    "username": " user name",
    "password": " password"
}</pre>
If using OAuth 2.0 authentication:
<pre>{
    "clientId": " client ID",
    "clientSecret": " client secret",
    "tokenUrl": " token URL"
}</pre></td></tr>
<tr><td>syncMode</td><td>Specify how Amazon Kendra should update your index when your data source content changes. You can choose between:
<code>FORCED_FULL_CRAWL</code> to freshly index all content, replacing existing content each time your data source syncs with your index.
<code>FULL_CRAWL</code> to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.</td></tr>
</table>

Configuration	Description
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.
version	The version of this template that is currently supported.

Alfresco JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "siteId": {
              "type": "string"
            },
            "repoUrl": {
              "type": "string"
            },
            "webAppUrl": {
              "type": "string"
            },
            "repositoryAdditionalProperties": {
              "type": "object",
              "properties": {
                "authType": {
```

```
                    "type": "string",
                    "enum": [
                      "OAuth2",
                      "Basic"
                    ]
                  },
                  "type": {
                    "type": "string",
                    "enum": [
                      "PAAS",
                      "ON_PREM"
                    ]
                  },
                  "crawlType": {
                    "type": "string",
                    "enum": [
                      "ASPECT",
                      "SITE_ID",
                      "ALL_SITES"
                    ]
                  }
                }
              }
            }
          }
        },
        "required": [
          "repositoryEndpointMetadata"
        ]
      },
      "repositoryConfigurations": {
        "type": "object",
        "properties": {
          "document": {
            "type": "object",
            "properties": {
              "fieldMappings": {
                "type": "array",
                "items": {
                  "anyOf": [
                    {
                      "type": "object",
                      "properties": {
                        "indexFieldName": {
```

```
                        "type": "string"
                      },
                      "indexFieldType": {
                        "type": "string",
                        "enum": [
                          "STRING",
                          "DATE",
                          "STRING_LIST",
                          "LONG"
                        ]
                      },
                      "dataSourceFieldName": {
                        "type": "string"
                      },
                      "dateFieldFormat": {
                        "type": "string",
                        "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                      }
                    },
                    "required": [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
                    ]
                  }
                ]
              }
            }
          },
          "required": [
            "fieldMappings"
          ]
        },
        "comment": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": {
                "anyOf": [
                  {
                    "type": "object",
                    "properties": {
                      "indexFieldName": {
```

```
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
                    "enum": [
                      "STRING",
                      "DATE",
                      "STRING_LIST",
                      "LONG"
                    ]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  },
                  "dateFieldFormat": {
                    "type": "string",
                    "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                  }
                },
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        }
      },
      "required": [
        "fieldMappings"
      ]
    }
  }
},
"additionalProperties": {
  "type": "object",
  "properties": {
    "aspectName": {
      "type": "string"
    },
    "aspectProperties": {
      "type": "array"
    },
```

```
          "enableFineGrainedControl": {
            "type": "boolean"
          },
          "isCrawlComment": {
            "type": "boolean"
          },
          "inclusionFileNamePatterns": {
            "type": "array"
          },
          "exclusionFileNamePatterns": {
            "type": "array"
          },
          "inclusionFileTypePatterns": {
            "type": "array"
          },
          "exclusionFileTypePatterns": {
            "type": "array"
          },
          "inclusionFilePathPatterns": {
            "type": "array"
          },
          "exclusionFilePathPatterns": {
            "type": "array"
          }
        }
      },
      "type": {
        "type": "string",
        "pattern": "ALFRESCO"
      },
      "secretArn": {
        "type": "string",
        "minLength": 20,
        "maxLength": 2048
      },
      "syncMode": {
        "type": "string",
        "enum": [
          "FORCED_FULL_CRAWL",
          "FULL_CRAWL"
        ]
      },
      "enableIdentityCrawler": {
        "type": "boolean"
```

```
    },
    "version": {
      "type": "string",
      "anyOf": [
        {
          "pattern": "1.0.0"
        }
      ]
    }
  },
  "required": [
    "connectionConfiguration",
    "repositoryConfigurations",
    "additionalProperties",
    "type",
    "secretArn"
  ]
}
```

Aurora (MySQL) template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as `JDBC`, the database type as `mysql`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Aurora (MySQL) JSON schema.

The following table describes the parameters of the Aurora (MySQL) JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	Required configuration information for connecting your data source. • dbType—The type of Java database that you use, whether `mysql`, `db2`, `postgresql` , `oracle`, or `sqlserver` .

Configuration	Description
	• dbHost—The database host name. • dbPort—The database port. • dbInstance—The database instance.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings. Specify the type of data source and the secret ARN.
document	A list of objects that map the attributes or field names of your database content to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.
primaryKey	Provide the primary key for the database table. This identifies a table within your database.
titleColumn	Provide the name of the document title column within your database table.
bodyColumn	Provide the name of the document title column within your database table.
sqlQuery	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.

Configuration	Description
timestampColumn	Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
timestampFormat	Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.
timezone	Enter the name of the column which contains time zones for the content to be crawled.
changeDetectingColumns	Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns
allowedUsersColumns	Enter the name of the column which contains User IDs to be allowed access to content.
allowedGroupsColumn	Enter the name of the column which contains User IDs to be allowed access to content.
sourceURIColumn	Enter the name of the column which contains Source URLs to be indexed.
isSslEnabled	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
type	The type of data source. Specify `JDBC` as your data source type.

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
secretArn	The Amazon Resource Name (ARN) of a Secrets Manager secret that contains user name and password required to connect to your database. The secret must contain a JSON structure with the following keys: `{ "user name": "database user name", "password": " password" }`
version	The version of the template that is currently supported.

Aurora (MySQL) JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "dbType": {
              "type": "string",
              "enum": [
                "mysql",
                "db2",
                "postgresql",
                "oracle",
                "sqlserver"
              ]
            },
            "dbHost": {
              "type": "string"
            },
            "dbPort": {
              "type": "string"
            },
            "dbInstance": {
              "type": "string"
            }
          },
          "required": [
            "dbType",
            "dbHost",
            "dbPort",
            "dbInstance"
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
```

```
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "document": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string"
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "primaryKey": {
          "type": "string"
```

```
        },
        "titleColumn": {
          "type": "string"
        },
        "bodyColumn": {
          "type": "string"
        },
        "sqlQuery": {
          "type": "string",
          "not": {
            "pattern": ";+"
          }
        },
        "timestampColumn": {
          "type": "string"
        },
        "timestampFormat": {
          "type": "string"
        },
        "timezone": {
          "type": "string"
        },
        "changeDetectingColumns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "allowedUsersColumn": {
          "type": "string"
        },
        "allowedGroupsColumn": {
          "type": "string"
        },
        "sourceURIColumn": {
          "type": "string"
        },
        "isSslEnabled": {
          "type": "boolean"
        }
      },
      "required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
    },
    "type" : {
```

```
        "type" : "string",
        "pattern": "JDBC"
      },
      "syncMode": {
        "type": "string",
        "enum": [
          "FORCED_FULL_CRAWL",
          "FULL_CRAWL",
          "CHANGE_LOG"
        ]
      },
      "secretArn": {
        "type": "string"
      }
    },
    "version": {
      "type": "string",
      "anyOf": [
        {
          "pattern": "1.0.0"
        }
      ]
    },
    "required": [
        "connectionConfiguration",
        "repositoryConfigurations",
        "syncMode",
        "additionalProperties",
        "secretArn",
        "type"
    ]
}
```

Aurora (PostgreSQL) template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as `JDBC`, the database type as `postgresql`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Aurora (PostgreSQL) JSON schema.

The following table describes the parameters of the Aurora (PostgreSQL) JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	Required configuration information for connecting your data source. • dbType—The type of Java database that you use, whether `mysql`, `db2`, `postgresql`, `oracle`, or `sqlserver`. • dbHost—The database host name. • dbPort—The database port. • dbInstance—The database instance.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings. Specify the type of data source and the secret ARN.
document	A list of objects that map the attributes or field names of your database content to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.
primaryKey	Provide the primary key for the database table. This identifies a table within your database.
titleColumn	Provide the name of the document title column within your database table.

Configuration	Description
bodyColumn	Provide the name of the document title column within your database table.
sqlQuery	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
timestampColumn	Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
timestampFormat	Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.
timezone	Enter the name of the column which contains time zones for the content to be crawled.
changeDetectingColumns	Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns
allowedUsersColumns	Enter the name of the column which contains User IDs to be allowed access to content.
allowedGroupsColumn	Enter the name of the column which contains User IDs to be allowed access to content.
sourceURIColumn	Enter the name of the column which contains Source URLs to be indexed.

Configuration	Description
isSslEnabled	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
type	The type of data source. Specify `JDBC` as your data source type.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

Configuration	Description
secretArn	The Amazon Resource Name (ARN) of a Secrets Manager secret that contains user name and password required to connect to your database. The secret must contain a JSON structure with the following keys: `{ "user name": "database user name", "password": " password" }`
version	The version of the template that is currently supported.

Aurora (PostgreSQL) JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "dbType": {
              "type": "string",
              "enum": [
                "mysql",
                "db2",
                "postgresql",
                "oracle",
                "sqlserver"
              ]
            },
            "dbHost": {
              "type": "string"
```

```
            },
            "dbPort": {
              "type": "string"
            },
            "dbInstance": {
              "type": "string"
            }
          },
          "required": [
            "dbType",
            "dbHost",
            "dbPort",
            "dbInstance"
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "document": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string"
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    }
                  },
                  "required": [
                    "indexFieldName",
```

```
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "primaryKey": {
          "type": "string"
        },
        "titleColumn": {
          "type": "string"
        },
        "bodyColumn": {
          "type": "string"
        },
        "sqlQuery": {
          "type": "string",
          "not": {
            "pattern": ";+"
          }
        },
        "timestampColumn": {
          "type": "string"
        },
        "timestampFormat": {
          "type": "string"
        },
        "timezone": {
          "type": "string"
        },
        "changeDetectingColumns": {
          "type": "array",
```

```
          "items": {
            "type": "string"
          }
        },
        "allowedUsersColumn": {
          "type": "string"
        },
        "allowedGroupsColumn": {
          "type": "string"
        },
        "sourceURIColumn": {
          "type": "string"
        },
        "isSslEnabled": {
          "type": "boolean"
        }
      },
      "required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
    },
    "type" : {
      "type" : "string",
      "pattern": "JDBC"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL",
        "CHANGE_LOG"
      ]
    },
    "secretArn": {
      "type": "string"
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
```

```
        "connectionConfiguration",
        "repositoryConfigurations",
        "syncMode",
        "additionalProperties",
        "secretArn",
        "type"
    ]
}
```

Amazon RDS (Microsoft SQL Server) template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as `JDBC`, the database type as `sqlserver`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Amazon RDS (Microsoft SQL Server) JSON schema.

The following table describes the parameters of the Amazon RDS (Microsoft SQL Server) JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	Required configuration information for connecting your data source. • dbType—The type of Java database that you use, whether `mysql`, `db2`, `postgresql` , `oracle`, or `sqlserver` . • dbHost—The database host name. • dbPort—The database port. • dbInstance—The database instance.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring

Configuration	Description
	specific types of content and field mappings. Specify the type of data source and the secret ARN.
document	A list of objects that map the attributes or field names of your database content to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.
primaryKey	Provide the primary key for the database table. This identifies a table within your database.
titleColumn	Provide the name of the document title column within your database table.
bodyColumn	Provide the name of the document title column within your database table.
sqlQuery	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
timestampColumn	Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
timestampFormat	Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

Configuration	Description
timezone	Enter the name of the column which contains time zones for the content to be crawled.
changeDetectingColumns	Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns
allowedUsersColumns	Enter the name of the column which contains User IDs to be allowed access to content.
allowedGroupsColumn	Enter the name of the column which contains User IDs to be allowed access to content.
sourceURIColumn	Enter the name of the column which contains Source URLs to be indexed.
isSslEnabled	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
type	The type of data source. Specify `JDBC` as your data source type.

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

secretArn	The Amazon Resource Name (ARN) of a Secrets Manager secret that contains user name and password required to connect to your database. The secret must contain a JSON structure with the following keys:
	`{`
	`    "user name": "database user name",`
	`    "password": " password"`
	`}`

version	The version of the template that is currently supported.

Amazon RDS (Microsoft SQL Server) JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "dbType": {
              "type": "string",
              "enum": [
                "mysql",
                "db2",
                "postgresql",
                "oracle",
                "sqlserver"
              ]
            },
            "dbHost": {
              "type": "string"
            },
            "dbPort": {
              "type": "string"
            },
            "dbInstance": {
              "type": "string"
            }
          },
          "required": [
            "dbType",
            "dbHost",
            "dbPort",
            "dbInstance"
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
```

```
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "document": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string"
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "primaryKey": {
          "type": "string"
```

```
        },
        "titleColumn": {
          "type": "string"
        },
        "bodyColumn": {
          "type": "string"
        },
        "sqlQuery": {
          "type": "string",
          "not": {
            "pattern": ";+"
          }
        },
        "timestampColumn": {
          "type": "string"
        },
        "timestampFormat": {
          "type": "string"
        },
        "timezone": {
          "type": "string"
        },
        "changeDetectingColumns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "allowedUsersColumn": {
          "type": "string"
        },
        "allowedGroupsColumn": {
          "type": "string"
        },
        "sourceURIColumn": {
          "type": "string"
        },
        "isSslEnabled": {
          "type": "boolean"
        }
      },
      "required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
    },
    "type" : {
```

```
      "type" : "string",
      "pattern": "JDBC"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL",
        "CHANGE_LOG"
      ]
    },
    "secretArn": {
      "type": "string"
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
      "connectionConfiguration",
      "repositoryConfigurations",
      "syncMode",
      "additionalProperties",
      "secretArn",
      "type"
  ]
}
```

Amazon RDS (MySQL) template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as `JDBC`, the database type as `mysql`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Amazon RDS (MySQL) JSON schema.

The following table describes the parameters of the Amazon RDS (MySQL) JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	Required configuration information for connecting your data source. • dbType—The type of Java database that you use, whether `mysql`, `db2`, `postgresql`, `oracle`, or `sqlserver`. • dbHost—The database host name. • dbPort—The database port. • dbInstance—The database instance.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings. Specify the type of data source and the secret ARN.
document	A list of objects that map the attributes or field names of your database content to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.
primaryKey	Provide the primary key for the database table. This identifies a table within your database.

Configuration	Description
titleColumn	Provide the name of the document title column within your database table.
bodyColumn	Provide the name of the document title column within your database table.
sqlQuery	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
timestampColumn	Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
timestampFormat	Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.
timezone	Enter the name of the column which contains time zones for the content to be crawled.
changeDetectingColumns	Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns
allowedUsersColumns	Enter the name of the column which contains User IDs to be allowed access to content.
allowedGroupsColumn	Enter the name of the column which contains User IDs to be allowed access to content.
sourceURIColumn	Enter the name of the column which contains Source URLs to be indexed.

Configuration	Description
isSslEnabled	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
type	The type of data source. Specify `JDBC` as your data source type.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

Configuration	Description
secretArn	The Amazon Resource Name (ARN) of a Secrets Manager secret that contains user name and password required to connect to your database. The secret must contain a JSON structure with the following keys: `{ "user name": "database user name", "password": " password" }`
version	The version of the template that is currently supported.

Amazon RDS (MySQL) JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "dbType": {
              "type": "string",
              "enum": [
                "mysql",
                "db2",
                "postgresql",
                "oracle",
                "sqlserver"
              ]
            },
            "dbHost": {
              "type": "string"
```

```
          },
          "dbPort": {
            "type": "string"
          },
          "dbInstance": {
            "type": "string"
          }
        },
        "required": [
          "dbType",
          "dbHost",
          "dbPort",
          "dbInstance"
        ]
      }
    },
    "required": [
      "repositoryEndpointMetadata"
    ]
  },
  "repositoryConfigurations": {
    "type": "object",
    "properties": {
      "document": {
        "type": "object",
        "properties": {
          "fieldMappings": {
            "type": "array",
            "items": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string"
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  }
                },
                "required": [
                  "indexFieldName",
```

```
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "primaryKey": {
          "type": "string"
        },
        "titleColumn": {
          "type": "string"
        },
        "bodyColumn": {
          "type": "string"
        },
        "sqlQuery": {
          "type": "string",
          "not": {
            "pattern": ";+"
          }
        },
        "timestampColumn": {
          "type": "string"
        },
        "timestampFormat": {
          "type": "string"
        },
        "timezone": {
          "type": "string"
        },
        "changeDetectingColumns": {
          "type": "array",
```

```
          "items": {
            "type": "string"
          }
        },
        "allowedUsersColumn": {
          "type": "string"
        },
        "allowedGroupsColumn": {
          "type": "string"
        },
        "sourceURIColumn": {
          "type": "string"
        },
        "isSslEnabled": {
          "type": "boolean"
        }
      },
      "required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
    },
    "type" : {
      "type" : "string",
      "pattern": "JDBC"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL",
        "CHANGE_LOG"
      ]
    },
    "secretArn": {
      "type": "string"
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
```

```
        "connectionConfiguration",
        "repositoryConfigurations",
        "syncMode",
        "additionalProperties",
        "secretArn",
        "type"
    ]
}
```

Amazon RDS (Oracle) template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as `JDBC`, the database type as `oracle`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Amazon RDS (Oracle) JSON schema.

The following table describes the parameters of the Amazon RDS (Oracle) JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	Required configuration information for connecting your data source. • dbType—The type of Java database that you use, whether `mysql`, `db2`, `postgresql`, `oracle`, or `sqlserver`. • dbHost—The database host name. • dbPort—The database port. • dbInstance—The database instance.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.

Configuration	Description
	Specify the type of data source and the secret ARN.
document	A list of objects that map the attributes or field names of your database content to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.
primaryKey	Provide the primary key for the database table. This identifies a table within your database.
titleColumn	Provide the name of the document title column within your database table.
bodyColumn	Provide the name of the document title column within your database table.
sqlQuery	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
timestampColumn	Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
timestampFormat	Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

Configuration	Description
timezone	Enter the name of the column which contains time zones for the content to be crawled.
changeDetectingColumns	Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns
allowedUsersColumns	Enter the name of the column which contains User IDs to be allowed access to content.
allowedGroupsColumn	Enter the name of the column which contains User IDs to be allowed access to content.
sourceURIColumn	Enter the name of the column which contains Source URLs to be indexed.
isSslEnabled	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
type	The type of data source. Specify `JDBC` as your data source type.

<table>
<tr><th>Configuration</th><th>Description</th></tr>
<tr><td>syncMode</td><td>Specify how Amazon Kendra should update your index when your data source content changes. You can choose between:
<code>FORCED_FULL_CRAWL</code> to freshly index all content, replacing existing content each time your data source syncs with your index.
<code>FULL_CRAWL</code> to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
<code>CHANGE_LOG</code> to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.</td></tr>
<tr><td>secretArn</td><td>The Amazon Resource Name (ARN) of a Secrets Manager secret that contains user name and password required to connect to your database. The secret must contain a JSON structure with the following keys:<pre><code>{
    "user name": "database user name",
    "password": " password"
}</code></pre></td></tr>
<tr><td>version</td><td>The version of the template that is currently supported.</td></tr>
</table>

Amazon RDS (Oracle) JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "dbType": {
              "type": "string",
              "enum": [
                "mysql",
                "db2",
                "postgresql",
                "oracle",
                "sqlserver"
              ]
            },
            "dbHost": {
              "type": "string"
            },
            "dbPort": {
              "type": "string"
            },
            "dbInstance": {
              "type": "string"
            }
          },
          "required": [
            "dbType",
            "dbHost",
            "dbPort",
            "dbInstance"
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
```

```
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "document": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string"
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "primaryKey": {
          "type": "string"
```

```
        },
        "titleColumn": {
          "type": "string"
        },
        "bodyColumn": {
          "type": "string"
        },
        "sqlQuery": {
          "type": "string",
          "not": {
            "pattern": ";+"
          }
        },
        "timestampColumn": {
          "type": "string"
        },
        "timestampFormat": {
          "type": "string"
        },
        "timezone": {
          "type": "string"
        },
        "changeDetectingColumns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "allowedUsersColumn": {
          "type": "string"
        },
        "allowedGroupsColumn": {
          "type": "string"
        },
        "sourceURIColumn": {
          "type": "string"
        },
        "isSslEnabled": {
          "type": "boolean"
        }
      },
      "required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
    },
    "type" : {
```

```
        "type" : "string",
        "pattern": "JDBC"
      },
      "syncMode": {
        "type": "string",
        "enum": [
          "FORCED_FULL_CRAWL",
          "FULL_CRAWL",
          "CHANGE_LOG"
        ]
      },
      "secretArn": {
        "type": "string"
      }
    },
    "version": {
      "type": "string",
      "anyOf": [
        {
          "pattern": "1.0.0"
        }
      ]
    },
    "required": [
        "connectionConfiguration",
        "repositoryConfigurations",
        "syncMode",
        "additionalProperties",
        "secretArn",
        "type"
    ]
  }
```

Amazon RDS (PostgreSQL) template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as `JDBC`, the database type as `postgresql`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Amazon RDS (PostgreSQL) JSON schema.

The following table describes the parameters of the Amazon RDS (PostgreSQL) JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	Required configuration information for connecting your data source. • dbType—The type of Java database that you use, whether `mysql`, `db2`, `postgresql` , `oracle`, or `sqlserver` . • dbHost—The database host name. • dbPort—The database port. • dbInstance—The database instance.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings. Specify the type of data source and the secret ARN.
document	A list of objects that map the attributes or field names of your database content to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.
primaryKey	Provide the primary key for the database table. This identifies a table within your database.

Configuration	Description
titleColumn	Provide the name of the document title column within your database table.
bodyColumn	Provide the name of the document title column within your database table.
sqlQuery	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
timestampColumn	Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
timestampFormat	Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.
timezone	Enter the name of the column which contains time zones for the content to be crawled.
changeDetectingColumns	Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns
allowedUsersColumns	Enter the name of the column which contains User IDs to be allowed access to content.
allowedGroupsColumn	Enter the name of the column which contains User IDs to be allowed access to content.
sourceURIColumn	Enter the name of the column which contains Source URLs to be indexed.

Configuration	Description
isSslEnabled	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
type	The type of data source. Specify `JDBC` as your data source type.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

Configuration	Description
secretArn	The Amazon Resource Name (ARN) of a Secrets Manager secret that contains user name and password required to connect to your database. The secret must contain a JSON structure with the following keys: `{ "user name": "database user name", "password": " password" }`
version	The version of the template that is currently supported.

Amazon RDS (PostgreSQL) JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "dbType": {
              "type": "string",
              "enum": [
                "mysql",
                "db2",
                "postgresql",
                "oracle",
                "sqlserver"
              ]
            },
            "dbHost": {
              "type": "string"
```

```
            },
            "dbPort": {
              "type": "string"
            },
            "dbInstance": {
              "type": "string"
            }
          },
          "required": [
            "dbType",
            "dbHost",
            "dbPort",
            "dbInstance"
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "document": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string"
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    }
                  },
                  "required": [
                    "indexFieldName",
```

```
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "primaryKey": {
          "type": "string"
        },
        "titleColumn": {
          "type": "string"
        },
        "bodyColumn": {
          "type": "string"
        },
        "sqlQuery": {
          "type": "string",
          "not": {
            "pattern": ";+"
          }
        },
        "timestampColumn": {
          "type": "string"
        },
        "timestampFormat": {
          "type": "string"
        },
        "timezone": {
          "type": "string"
        },
        "changeDetectingColumns": {
          "type": "array",
```

```
          "items": {
            "type": "string"
          }
        },
        "allowedUsersColumn": {
          "type": "string"
        },
        "allowedGroupsColumn": {
          "type": "string"
        },
        "sourceURIColumn": {
          "type": "string"
        },
        "isSslEnabled": {
          "type": "boolean"
        }
      },
      "required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
    },
    "type" : {
      "type" : "string",
      "pattern": "JDBC"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL",
        "CHANGE_LOG"
      ]
    },
    "secretArn": {
      "type": "string"
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
```

```
        "connectionConfiguration",
        "repositoryConfigurations",
        "syncMode",
        "additionalProperties",
        "secretArn",
        "type"
    ]
}
```

Amazon S3 template schema

You include a JSON that contains the data source schema as part of the template configuration. You provide the name of the S3 bucket as a part of the connection configuration or repository endpoint details. Also specify the type of data source as S3, and other necessary configurations. You then specify TEMPLATE as the Type when you call CreateDataSource.

You can use the template provided in this developer guide. See S3 JSON schema.

The following table describes the parameters of the Amazon S3 JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
BucketName	The name of your Amazon S3 bucket.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
additionalProperties	Additional configuration options for your content in your data source
• inclusionPatterns • exclusionPatterns • inclusionPrefixes • exclusionPrefixes	A list of regular expression patterns to include or exclude specific files in your Amazon S3 data source. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a

Configuration	Description
	file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedenc e and the file isn't included in the index.
aclConfigurationFilePath	The file path that controls access to documents in an Amazon Kendra index.
metadataFilesPrefix	The location within your bucket for metadata files.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
type	The type of data source. Specify S3 as your data source type.
version	The version of the template that is supported.

S3 JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
```

```
        "type": "object",
        "properties": {
          "repositoryEndpointMetadata": {
            "type": "object",
            "properties": {
              "BucketName": {
                "type": "string"
              }
            },
            "required": [
              "BucketName"
            ]
          }
        },
        "required": [
          "repositoryEndpointMetadata"
        ]
      },
      "repositoryConfigurations": {
        "type": "object",
        "properties": {
          "document": {
            "type": "object",
            "properties": {
              "fieldMappings": {
                "type": "array",
                "items": [
                  {
                    "type": "object",
                    "properties": {
                      "indexFieldName": {
                        "type": "string"
                      },
                      "indexFieldType": {
                        "type": "string",
                        "enum": [
                          "STRING"
                        ]
                      },
                      "dataSourceFieldName": {
                        "type": "string"
                      }
                    },
                    "required": [
```

```
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
        "document"
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "inclusionPatterns": {
          "type": "array"
        },
        "exclusionPatterns": {
          "type": "array"
        },
        "inclusionPrefixes": {
          "type": "array"
        },
        "exclusionPrefixes": {
          "type": "array"
        },
        "aclConfigurationFilePath": {
          "type": "string"
        },
        "metadataFilesPrefix": {
          "type": "string"
        }
      }
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FULL_CRAWL",
```

```
          "FORCED_FULL_CRAWL"
        ]
      },
      "type": {
        "type": "string",
        "pattern": "S3"
      },
      "version": {
        "type": "string",
        "anyOf": [
          {
            "pattern": "1.0.0"
          }
        ]
      }
    },
    "required": [
      "connectionConfiguration",
      "type",
      "syncMode",
      "repositoryConfigurations"
    ]
  }
```

Amazon Kendra Web Crawler template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object.

You provide the seed or starting point URLs, or you can provide the sitemap URLs, as part of the connection configuration or repository endpoint details. Instead of manually listing all your URLs, you can provide the path to the Amazon S3 bucket that stores a text file for your list of seed URLs or sitemap XML files, which you can club together in a ZIP file in S3.

You also specify the type of data source as `WEBCRAWLERV2`, the website authentication credentials and authentication type if your websites require authentication, and other necessary configurations.

You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

> **Important**
>
> Web Crawler v2.0 connector creation is not supported by AWS CloudFormation. Use the Web Crawler v1.0 connector if you need AWS CloudFormation support.

When selecting websites to index, you must adhere to the Amazon Acceptable Use Policy and all other Amazon terms. Remember that you must only use Amazon Kendra Web Crawler to index your own web pages, or web pages that you have authorization to index. To learn how to stop Amazon Kendra Web Crawler from indexing your websites, see Configuring the `robots.txt` file for Amazon Kendra Web Crawler.

You can use the template provided in this developer guide. See Amazon Kendra Web Crawler JSON schema.

The following table describes the parameters of the Amazon Kendra Web Crawler JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
siteMapUrls	The list of sitemap URLs for the websites that you want to crawl. You can list up to three sitemap URLs.
s3SeedUrl	The S3 path to the text file that stores the list of seed or starting point URLs. For example, *s3://bucket-name/directory/*. Each URL in the text file must be formatted on a separate line. You can list up to 100 seed URLs in a file.
s3SiteMapUrl	The S3 path to the sitemap XML files. For example, *s3://bucket-name/directory/*. You can list up to three sitemap XML files. You can club together multiple sitemap files into a ZIP

Configuration	Description
	file and store the ZIP file in your Amazon S3 bucket.
seedUrlConnections	The list of seed or starting point URLs for the websites that you want to crawl.You can list up to 100 seed URLs.
seedUrl	The seed or starting point URL.
authentication	The authentication type if your websites require the same authentication, otherwise specify `NoAuthentication` .
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
• webPage • attachment	A list of objects that map the attributes or field names of your web pages and web page files to Amazon Kendra index field names. For example, the HTML web page title tag can be mapped to the `_document_title` index field. For more information, see [Mapping data source fields]().

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
additionalProperties	Additional configuration options for your content in your data source.
rateLimit	The maximum number of URLs crawled per website host per minute.
maxFileSize	The maximum size (in MB) of a web page or attachment to crawl.
crawlDepth	The number of levels from the seed URL to crawl. For example, the seed URL page is depth 1 and any hyperlinks on this page that are also crawled are depth 2.
maxLinksPerUrl	The maximum number of URLs on a web page to include when crawling a website. This number is per web page. As a website's web pages are crawled, any URLs that the webpages link to also are crawled. URLs on a web page are crawled in order of appearance.

Configuration	Description
crawlSubDomain	`true` to crawl the website domains with subdomains. For example, if the seed URL is "abc.example.com", then "a.abc.example.com " and "b.abc.example.com" are also crawled. If you don't set `crawlSubDomain` or `crawlAllDomain` to `true`, then Amazon Kendra only crawls the domains of the websites that you want to crawl.
crawlAllDomain	`true` to crawl the website domains with subdomains and other domains the web pages link to. If you don't set `crawlSubDomain` or `crawlAllDomain` to `true`, then Amazon Kendra only crawls the domains of the websites that you want to crawl.
honorRobots	`true` to respect the robots.txt directives of the websites that you want to crawl. These directives control how Amazon Kendra Web Crawler crawls the websites, whether Amazon Kendra can crawl only specific content or not crawl any content.
crawlAttachments	`true` to crawl files that the web pages link to.
• inclusionURLCrawlPatterns • inclusionURLIndexPatterns	A list of regular expression patterns to *include* crawling certain URLs and indexing any hyperlinks on these URL web pages. URLs that match the patterns are included in the index. URLs that don't match the patterns are excluded from the index. If a URL matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the URL/website's web pages aren't included in the index.

Configuration	Description
• exclusionURLCrawlPatterns • exclusionURLIndexPatterns	A list of regular expression patterns to *exclude* crawling certain URLs and indexing any hyperlinks on these URL web pages. URLs that match the patterns are excluded from the index. URLs that don't match the patterns are included in the index. If a URL matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the URL/website's web pages aren't included in the index.
inclusionFileIndexPatterns	A list of regular expression patterns to *include* certain web page files. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the file isn't included in the index.
exclusionFileIndexPatterns	A list of regular expression patterns to *exclude* certain web page files. Files that match the patterns are excluded from the index. Files that don't match the patterns are included in the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the file isn't included in the index.
implicitWaitDuration	implicitWaitDuration specifies how long the connector will wait, in seconds, before crawling a webpage. Range: 0-10 eg. "implicitWaitDuration": "5"

Configuration	Description
proxy	Configuration information required to connect to your internal websites via a web proxy.
host	The host name of the proxy sever you want to use to connect to internal websites. For example, the host name of *https://a.example.com/page1.html* is "a.example.com".
port	The port number of the proxy sever you want to use to connect to internal websites. For example, 443 is the standard port for HTTPS.
secretArn (proxy)	If web proxy credentials are required to connect to a website host, you can create an AWS Secrets Manager secret that stores the credentials. Provide the Amazon Resource Name (ARN) of the secret.
type	The type of data source. Specify `WEBCRAWLERV2` as your data source type.

Configuration	Description
secretArn	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that's used if your websites require authentication to access the websites. You store the authentication credentials for the website in the secret that contains JSON key-value pairs. If you use basic, or NTML/Kerberos, enter the user name and password. The JSON keys in the secret must be `userName` and `password`. NTLM authentication protocol includes password hashing, and Kerberos authentication protocol includes password encryption. If you use SAML or form authentication, enter the user name and password, XPath for the user name field (and user name button if using SAML), XPaths for the password field and button, and the login page URL. The JSON keys in the secret must be `userName`, `password`, `userNameFieldXpath`, `userNameButtonXpath`, `passwordFieldXpath`, `passwordButtonXpath`, and `loginPageUrl`. You can find the XPaths (XML Path Language) of elements using your web browser's developer tools. XPaths usually follow this format: `//tagname[@Attribute='Value']`. Amazon Kendra also checks if the endpoint information (seed URLs) included in the secret is the same the endpoint information specified in your data source endpoint configuration details.

Configuration	Description
version	The version of this template that is currently supported.

Amazon Kendra Web Crawler JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "siteMapUrls": {
              "type": "array",
              "items":{
                "type": "string",
                "pattern": "https://.*"
              }
            },
            "s3SeedUrl": {
              "type": "string",
              "pattern": "s3:.*"
            },
            "s3SiteMapUrl": {
              "type": "string",
              "pattern": "s3:.*"
            },
            "seedUrlConnections": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "seedUrl":{
                      "type": "string",
                      "pattern": "https://.*"
                    }
```

```
                },
                "required": [
                  "seedUrl"
                ]
              }
            ]
          },
          "authentication": {
            "type": "string",
            "enum": [
              "NoAuthentication",
              "BasicAuth",
              "NTLM_Kerberos",
              "Form",
              "SAML"
            ]
          }
        }
      }
    },
    "required": [
      "repositoryEndpointMetadata"
    ]
  },
  "repositoryConfigurations": {
    "type": "object",
    "properties": {
      "webPage": {
        "type": "object",
        "properties": {
          "fieldMappings": {
            "type": "array",
            "items": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
                    "enum": [
                      "STRING",
                      "DATE",
```

```
                        "LONG"
                      ]
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        },
        "attachment": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string",
                      "enum": [
                        "STRING",
                        "DATE",
                        "LONG"
                      ]
                    },
```

```
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      }
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL"
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "rateLimit": {
          "type": "string",
          "default": "300"
        },
        "maxFileSize": {
          "type": "string",
          "default": "50"
        },
        "crawlDepth": {
          "type": "string",
          "default": "2"
```

```
        },
        "maxLinksPerUrl": {
          "type": "string",
          "default": "100"
        },
        "crawlSubDomain": {
          "type": "boolean",
          "default": false
        },
        "crawlAllDomain": {
          "type": "boolean",
          "default": false
        },
        "honorRobots": {
          "type": "boolean",
          "default": false
        },
        "crawlAttachments": {
          "type": "boolean",
          "default": false
        },
        "inclusionURLCrawlPatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionURLCrawlPatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionURLIndexPatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionURLIndexPatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
```

```
        },
        "inclusionFileIndexPatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionFileIndexPatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "proxy": {
          "type": "object",
          "properties": {
            "host": {
              "type": "string"
            },
            "port": {
              "type": "string"
            },
            "secretArn": {
              "type": "string",
              "minLength": 20,
              "maxLength": 2048
            }
          }
        }
      },
      "implicitWaitDuration":  {
          "type":"object",
          "properties": {
            "innerNumber" : {
              "type": "number",
              "minimum": 0,
              "maximum": 10
            }
          }
        },
      "required": [
        "rateLimit",
        "maxFileSize",
        "crawlDepth",
```

```
          "crawlSubDomain",
          "crawlAllDomain",
          "maxLinksPerUrl",
          "honorRobots"
        ]
      },
      "type": {
        "type": "string",
        "pattern": "WEBCRAWLERV2"
      },
      "secretArn": {
        "type": "string",
        "minLength": 20,
        "maxLength": 2048
      }
    },
    "version": {
      "type": "string",
      "anyOf": [
        {
          "pattern": "1.0.0"
        }
      ]
    },
    "required": [
      "connectionConfiguration",
      "repositoryConfigurations",
      "syncMode",
      "type",
      "additionalProperties"
    ]
  }
```

Confluence template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the Confluence host URL, the hosting method, and the authentication type as a part of the connection configuration or repository endpoint details. Also specify the type of data source as `CONFLUENCEV2`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Confluence JSON schema.

The following table describes the parameters of the Confluence JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
hostUrl	The URL for your Confluence instance. For example, *`https://example.confluence.com`* .
type	The hosting method for your Confluence instance, whether `SAAS` and `ON_PREM`.
authType	The authentication method for your Confluence instance, whether `Basic`, `OAuth2`, or `Personal-token` .
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
• space • page • blog • comment • attachment	A list of objects that map the attributes or field names of your Confluence spaces, pages, blogs, comments, and attachments to Amazon Kendra index field names. For more information, see [Mapping data source fields](). The Confluence data source field names must exist in your Confluence custom metadata.
additionalProperties	Additional configuration options for your content in your data source.
isCrawlAcl	Configure `true` to crawl the access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. Note that the ACL specifies which

Configuration	Description
	documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. This means that if `isCrawlACL` is turned off, documents can be publicly searched. For more information, see [User context filtering]().
fieldForUserId	Specify `email` if you want to use the user email for the user ID. `email` is used by default and is currently the only supported user ID type.
• inclusionSpaceKeyFilter • exclusionSpaceKeyFilter • pageTitleRegEX • blogTitleRegEX • commentTitleRegEX • attachmentTitleRegEX • inclusionFileTypePatterns • exclusionFileTypePatterns • inclusionUrlPatterns • exclusionUrlPatterns	A list of regular expression patterns to include and/or exclude certain files in your Confluence data source. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.
proxyHost	The host name of the web proxy that you use, without the `http://` or `https://` protocol.
proxyPort	The port number used by the host URL transport protocol. Must be a numeric value between 0 and 65535.

Configuration	Description
• isCrawlPersonalSpace • isCrawlArchivedSpace • isCrawlArchivedPage • isCrawlPage • isCrawlBlog • isCrawlPageComment • isCrawlPageAttachment • isCrawlBlogComment • isCrawlBlogAttachment	`true` to crawl files in your Confluenc e personal spaces, pages, blogs, page comments, page attachments, blog comments, and blog attachments.
maxFileSizeInMegaBytes	Specify the file size limit in MBs that Amazon Kendra can crawl. Amazon Kendra crawls only the files within the size limit you define. The default file size is 50MB. The maximum file size should be greater than 0MB and less than or equal to 50MB.
type	The type of data source. Specify `CONFLUENCEV2` as your data source type.
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
secretARN	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Confluence. For information on these key-value pairs, see [Connection instructions for Confluence]().
version	The version of this template that is currently supported.

Confluence JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
```

```
          "properties": {
            "hostUrl": {
              "type": "string",
              "pattern": "https:.*"
            },
            "type": {
              "type": "string",
              "enum": [
                "SAAS",
                "ON_PREM"
              ]
            },
            "authType": {
              "type": "string",
              "enum": [
                "Basic",
                "OAuth2",
                "Personal-token"
              ]
            }
          },
          "required": [
            "hostUrl",
            "type",
            "authType"
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "space": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
```

```
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string",
                      "enum": [
                        "STRING",
                        "STRING_LIST",
                        "DATE"
                      ]
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        },
        "page": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
```

```
                  "indexFieldType": {
                    "type": "string",
                    "enum": [
                      "STRING",
                      "STRING_LIST",
                      "DATE",
                      "LONG"
                    ]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  },
                  "dateFieldFormat": {
                    "type": "string",
                    "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                  }
                },
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        },
        "required": [
          "fieldMappings"
        ]
      },
      "blog": {
        "type": "object",
        "properties": {
          "fieldMappings": {
            "type": "array",
            "items": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
```

```
                "enum": [
                  "STRING",
                  "STRING_LIST",
                  "DATE",
                  "LONG"
                ]
              },
              "dataSourceFieldName": {
                "type": "string"
              },
              "dateFieldFormat": {
                "type": "string",
                "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
              }
            },
            "required": [
              "indexFieldName",
              "indexFieldType",
              "dataSourceFieldName"
            ]
          }
        ]
      }
    },
    "required": [
      "fieldMappings"
    ]
  },
  "comment": {
    "type": "object",
    "properties": {
      "fieldMappings": {
        "type": "array",
        "items": [
          {
            "type": "object",
            "properties": {
              "indexFieldName": {
                "type": "string"
              },
              "indexFieldType": {
                "type": "string",
                "enum": [
                  "STRING",
```

```
                    "STRING_LIST",
                    "DATE",
                    "LONG"
                  ]
                },
                "dataSourceFieldName": {
                  "type": "string"
                },
                "dateFieldFormat": {
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                }
              },
              "required": [
                "indexFieldName",
                "indexFieldType",
                "dataSourceFieldName"
              ]
            }
          ]
        }
      },
      "required": [
        "fieldMappings"
      ]
    },
    "attachment": {
      "type": "object",
      "properties": {
        "fieldMappings": {
          "type": "array",
          "items": [
            {
              "type": "object",
              "properties": {
                "indexFieldName": {
                  "type": "string"
                },
                "indexFieldType": {
                  "type": "string",
                  "enum": [
                    "STRING",
                    "STRING_LIST",
                    "DATE",
```

```
                        "LONG"
                      ]
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      }
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "usersAclS3FilePath": {
          "type": "string"
        },
        "isCrawlAcl": {
          "type": "boolean"
        },
        "fieldForUserId": {
          "type": "string"
        },
        "inclusionSpaceKeyFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
```

```
        "exclusionSpaceKeyFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "pageTitleRegEX": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "blogTitleRegEX": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "commentTitleRegEX": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "attachmentTitleRegEX": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "isCrawlPersonalSpace": {
          "type": "boolean"
        },
        "isCrawlArchivedSpace": {
          "type": "boolean"
        },
        "isCrawlArchivedPage": {
          "type": "boolean"
        },
        "isCrawlPage": {
          "type": "boolean"
        },
        "isCrawlBlog": {
          "type": "boolean"
```

```
        },
        "isCrawlPageComment": {
          "type": "boolean"
        },
        "isCrawlPageAttachment": {
          "type": "boolean"
        },
        "isCrawlBlogComment": {
          "type": "boolean"
        },
        "isCrawlBlogAttachment": {
          "type": "boolean"
        },
        "maxFileSizeInMegaBytes":  {
          "type":"string"
        },
        "inclusionFileTypePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionFileTypePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionUrlPatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionUrlPatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "proxyHost": {
          "type": "string"
        },
        "proxyPort": {
```

```
          "type": "string"
        }
      },
      "required": []
    },
    "type": {
      "type": "string",
      "pattern": "CONFLUENCEV2"
    },
    "enableIdentityCrawler": {
      "type": "boolean"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FULL_CRAWL",
        "FORCED_FULL_CRAWL"
      ]
    },
    "secretArn": {
      "type": "string",
      "minLength": 20,
      "maxLength": 2048
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
    "connectionConfiguration",
    "repositoryConfigurations",
    "syncMode",
    "additionalProperties",
    "secretArn",
    "type"
  ]
}
```

Dropbox template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the Dropbox app key, app secret, and access token as part of your secret that stores your authentication credentials. Also specify the type of data source as `DROPBOX`, the type of access token you want to use (temporary or permanent), and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Dropbox JSON schema.

The following table describes the parameters of the Dropbox JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source. This data source does not specify an endpoint in `repositoryEndpointMetadata` . Rather, the connection information is included in an AWS Secrets Manager secret that you provide the `secretArn` .
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
• file • paper • papert • shortcut	A list of objects that map the attributes or field names of your Dropbox files, Dropbox Paper, and shortcuts to Amazon Kendra index field names. For more information, see Mapping data source fields.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between:

Configuration	Description
	• `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.

Configuration	Description

secretARN	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Dropbox. The secret must contain a JSON structure with the following keys:
	`{`
	`    "appKey": "Dropbox app key",`
	`    "appSecret": " Dropbox app secret",`
	`    "accesstoken": " temporary access token or refresh access token"`
	`}`

additionalProperties	Additional configuration options for your content in your data source.
isCrawlAcl	`true` to crawl the access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
• inclusionFileNamePatterns • inclusionFileTypePatterns	A list of regular expression patterns to *include* certain file names and types in your Dropbox data source. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedenc e and the file isn't included in the index.

Configuration	Description
• exclusionFileNamePatterns • exclusionFileTypePatterns	A list of regular expression patterns to *exclude* certain file names and types in your Dropbox data source. Files that match the patterns are excluded from the index. Files that don't match the patterns are included in the index. If a file matches both an exclusion and inclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.
• crawlFile • crawlPaper • crawlPapert • crawlShortcut	`true` to crawl files in your Dropbox, Dropbox Paper documents, Dropbox Paper templates , and web page shortcuts stored in your Dropbox.
type	The type of data source. Specify `DROPBOX` as your data source type.
tokenType	Specify your access token type: permanent or temporary access token. It's recommend ed that you create a refresh access token that never expires in Dropbox rather that relying on a one-time access token that expires after 4 hours. You create an app and a refresh access token in the Dropbox developer console and provide the access token in your secret.
version	The version of this template that is currently supported.

Dropbox JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
```

```
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
          }
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "file": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": {
                "anyOf": [
                  {
                    "type": "object",
                    "properties": {
                      "indexFieldName": {
                        "type": "string"
                      },
                      "indexFieldType": {
                        "type": "string",
                        "enum": [
                          "STRING",
                          "STRING_LIST",
                          "LONG",
                          "DATE"
                        ]
                      },
                      "dataSourceFieldName": {
                        "type": "string"
                      },
                      "dateFieldFormat": {
                        "type": "string",
```

```
                  "pattern": "dd-MM-yyyy HH:mm:ss"
                }
              },
              "required": [
                "indexFieldName",
                "indexFieldType",
                "dataSourceFieldName"
              ]
            }
          ]
        }
      }
    },
    "required": [
      "fieldMappings"
    ]
  },
  "paper": {
    "type": "object",
    "properties": {
      "fieldMappings": {
        "type": "array",
        "items": {
          "anyOf": [
            {
              "type": "object",
              "properties": {
                "indexFieldName": {
                  "type": "string"
                },
                "indexFieldType": {
                  "type": "string",
                  "enum": [
                    "STRING",
                    "STRING_LIST",
                    "LONG",
                    "DATE"
                  ]
                },
                "dataSourceFieldName": {
                  "type": "string"
                },
                "dateFieldFormat": {
                  "type": "string",
```

```
                    "pattern": "dd-MM-yyyy HH:mm:ss"
                  }
                },
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        }
      },
      "required": [
        "fieldMappings"
      ]
    },
    "papert": {
      "type": "object",
      "properties": {
        "fieldMappings": {
          "type": "array",
          "items": {
            "anyOf": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
                    "enum": [
                      "STRING",
                      "STRING_LIST",
                      "LONG",
                      "DATE"
                    ]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  },
                  "dateFieldFormat": {
                    "type": "string",
```

```
                    "pattern": "dd-MM-yyyy HH:mm:ss"
                  }
                },
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        }
      },
      "required": [
        "fieldMappings"
      ]
    },
    "shortcut": {
      "type": "object",
      "properties": {
        "fieldMappings": {
          "type": "array",
          "items": {
            "anyOf": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
                    "enum": [
                      "STRING",
                      "STRING_LIST",
                      "LONG",
                      "DATE"
                    ]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  },
                  "dateFieldFormat": {
                    "type": "string",
```

```
                        "pattern": "dd-MM-yyyy HH:mm:ss"
                      }
                    },
                    "required": [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
                    ]
                  }
                ]
              }
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      }
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FULL_CRAWL",
        "FORCED_FULL_CRAWL",
        "CHANGE_LOG"
      ]
    },
    "enableIdentityCrawler": {
      "type": "boolean"
    },
    "secretArn": {
      "type": "string"
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "isCrawlAcl": {
          "type": "boolean"
        },
        "inclusionFileNamePatterns": {
          "type": "array"
        },
        "exclusionFileNamePatterns": {
          "type": "array"
```

```
        },
        "inclusionFileTypePatterns": {
          "type": "array"
        },
        "exclusionFileTypePatterns": {
          "type": "array"
        },
        "crawlFile": {
          "type": "boolean"
        },
        "crawlPaper": {
          "type": "boolean"
        },
        "crawlPapert": {
          "type": "boolean"
        },
        "crawlShortcut": {
          "type": "boolean"
        }
      }
    },
    "type": {
      "type": "string",
      "pattern": "DROPBOX"
    },
    "tokenType": {
      "type": "string",
      "enum": [
        "PERMANENT",
        "TEMPORARY"
      ]
    },
    "version": {
      "type": "string",
      "anyOf": [
        {
          "pattern": "1.0.0"
        }
      ]
    }
  },
  "additionalProperties": false,
  "required": [
    "connectionConfiguration",
```

```
    "repositoryConfigurations",
    "additionalProperties",
    "syncMode",
    "enableIdentityCrawler",
    "secretArn",
    "type",
    "tokenType"
  ]
}
```

Drupal template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the Drupal host URL and the authentication type as part of the connection configuration or repository endpoint details. Also specify the type of data source as DRUPAL, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Drupal JSON schema.

The following table describes the parameters of the Drupal JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
hostUrl	The host url of your Drupal website. For example, *https://<hostname>/<drupalsitename>* .
repositoryConfigurations	Configuration information for the content of the data source.
• content • comment • attachment	A list of objects that map the attributes or field names of your Drupal files. For more information, see Mapping data source fields.

Configuration	Description
	The Drupal data source field names must exist in your Drupal custom metadata.
additionalProperties	Additional configuration options for your content in your data source.
• inclusionFileNamePatterns • articleTitleInclusionPatterns • pageTitleInclusionPatterns • customContentTitleInclusionPatterns • basicBlockTitleInclusionPatterns • customBlockTitleInclusionPatterns	A list of regular expression patterns to *include* certain files in your Drupal data source. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.
• exclusionFileNamePatterns • articleTitleExclusionPatterns • pageTitleExclusionPatterns • customContentTitleExclusionPatterns • basicBlockTitleExclusionPatterns • customBlockTitleExclusionPatterns	A list of regular expression patterns to *exclude* certain files in your Drupal data source. Files that match the patterns are excluded from the index. Files that don't match the patterns are included in the index. If a file matches both an exclusion and inclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.
contentDefinitions • contentType • fieldDefinition • isCrawlComments • isCrawlFiles • isCrawlArticle • isCrawlBasicPage • isCrawlBasicBlock • isCrawlCustomContentTypesList	Specify the content types to crawl and whether to crawl comments and attachments for your selected content types.

Configuration	Description
type	The type of data source. Specify DRUPAL as your data source type.
authType	The type of authentication that you use, whether BASIC-AUTH or OAUTH2.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • FORCED_FULL_CRAWL to freshly index all content, replacing existing content each time your data source syncs with your index. • FULL_CRAWL to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • CHANGE_LOG to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

Configuration	Description
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.

secretARN	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Drupal. The secret must contain a JSON structure with the following keys:
	
	If using basic authentication:
	
	`{`
	`    "username":  "user name",`
	`    "passwords":  "password"`
	`}`
	
	If using OAuth 2.0 authentication:
	
	`{`
	`    "username":  "user name",`
	`    "password":  "password" ,`
	`    "clientId":  "client id",`
	`    "clientSecret":  "client secret"`
	`}`

version	The version of this template that is currently supported.

Drupal JSON schema

```
{
 "$schema": "http://json-schema.org/draft-04/schema#",
 "type": "object",
 "properties": {
  "connectionConfiguration": {
   "type": "object",
   "properties": {
    "repositoryEndpointMetadata": {
     "type": "object",
     "properties": {
      "hostUrl": {
       "type": "string",
       "pattern": "https:.*"
      }
     },
     "required": [
      "hostUrl"
     ]
    }
   },
   "required": [
    "repositoryEndpointMetadata"
   ]
  },
  "repositoryConfigurations": {
   "type": "object",
   "properties": {
    "content": {
     "type": "object",
     "properties": {
      "fieldMappings": {
       "type": "array",
       "items": [
        {
         "type": "object",
         "properties": {
          "indexFieldName": {
           "type": "string"
          },
          "indexFieldType": {
           "type": "string",
           "enum": [
```

```
            "STRING",
            "DATE"
           ]
          },
          "dataSourceFieldName": {
           "type": "string"
          },
          "dateFieldFormat": {
           "type": "string",
           "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
          }
         },
         "required": [
          "indexFieldName",
          "indexFieldType",
          "dataSourceFieldName"
         ]
        }
       ]
      }
     },
     "required": [
      "fieldMappings"
     ]
    },
    "comment": {
     "type": "object",
     "properties": {
      "fieldMappings": {
       "type": "array",
       "items": [
        {
         "type": "object",
         "properties": {
          "indexFieldName": {
           "type": "string"
          },
          "indexFieldType": {
           "type": "string",
           "enum": [
            "STRING",
            "DATE"
           ]
          },
```

```
          "dataSourceFieldName": {
           "type": "string"
          },
          "dateFieldFormat": {
           "type": "string",
           "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
          }
         },
         "required": [
          "indexFieldName",
          "indexFieldType",
          "dataSourceFieldName"
         ]
        }
       ]
      }
     },
     "required": [
      "fieldMappings"
     ]
    },
    "attachment": {
     "type": "object",
     "properties": {
      "fieldMappings": {
       "type": "array",
       "items": [
        {
         "type": "object",
         "properties": {
          "indexFieldName": {
           "type": "string"
          },
          "indexFieldType": {
           "type": "string",
           "enum": [
            "STRING",
            "DATE"
           ]
          },
          "dataSourceFieldName": {
           "type": "string"
          },
          "dateFieldFormat": {
```

```
            "type": "string",
            "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
           }
          },
          "required": [
           "indexFieldName",
           "indexFieldType",
           "dataSourceFieldName"
          ]
         }
        ]
       }
      },
      "required": [
       "fieldMappings"
      ]
     }
    }
   },
   "additionalProperties": {
    "type": "object",
    "properties": {
     "isCrawlArticle": {
      "type": "boolean"
     },
     "isCrawlBasicPage": {
      "type": "boolean"
     },
     "isCrawlBasicBlock": {
      "type": "boolean"
     },
     "crawlCustomContentTypesList": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "crawlCustomBlockTypesList": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "filePath": {
```

```
      "anyOf": [
       {
        "type": "string",
        "pattern": "s3:.*"
       },
       {
        "type": "string",
        "pattern": ""
       }
      ]
     },
     "inclusionFileNamePatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "exclusionFileNamePatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "articleTitleInclusionPatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "articleTitleExclusionPatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "pageTitleInclusionPatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "pageTitleExclusionPatterns": {
      "type": "array",
      "items": {
```

```
       "type": "string"
      }
     },
     "customContentTitleInclusionPatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "customContentTitleExclusionPatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "basicBlockTitleInclusionPatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "basicBlockTitleExclusionPatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "customBlockTitleInclusionPatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "customBlockTitleExclusionPatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "contentDefinitions": {
      "type": "array",
      "items": {
       "properties": {
        "contentType": {
```

```
          "type": "string"
        },
        "fieldDefinition": {
          "type": "array",
          "items": [
            {
              "type": "object",
              "properties": {
                "machineName": {
                  "type": "string"
                },
                "type": {
                  "type": "string"
                }
              },
              "required": [
                "machineName",
                "type"
              ]
            }
          ]
        },
        "isCrawlComments": {
          "type": "boolean"
        },
        "isCrawlFiles": {
          "type": "boolean"
        }
      }
    },
    "required": [
      "contentType",
      "fieldDefinition",
      "isCrawlComments",
      "isCrawlFiles"
    ]
  }
 },
 "required": []
},
"type": {
 "type": "string",
 "pattern": "DRUPAL"
},
```

```
   "authType": {
    "type": "string",
    "enum": [
     "BASIC-AUTH",
     "OAUTH2"
    ]
   },
   "syncMode": {
    "type": "string",
    "enum": [
     "FORCED_FULL_CRAWL",
     "FULL_CRAWL",
     "CHANGE_LOG"
    ]
   },
   "enableIdentityCrawler": {
    "type": "boolean"
   },
   "secretArn": {
    "type": "string",
    "minLength": 20,
    "maxLength": 2048
   }
  },
  "version": {
   "type": "string",
   "anyOf": [
    {
     "pattern": "1.0.0"
    }
   ]
  },
  "required": [
   "connectionConfiguration",
   "repositoryConfigurations",
   "syncMode",
   "additionalProperties",
   "secretArn",
   "type"
  ]
 }
```

GitHub template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the GitHub host URL, the organization name, and whether you use GitHub cloud or GitHub on-premises as part of the connection configuration or repository endpoint details. Also specify the type of data source as `GITHUB`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See GitHub JSON schema.

The following table describes the parameters of the GitHub JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
type	Specify the type as either `SAAS` or `ON_PREMISE`.
hostUrl	The GitHub host URL. For example, if you use GitHub SaaS/Enterprise Cloud: *https://api.github.com*. Or, if you use GitHub on-premises/Enterprise Server: *https://on-prem-host-url/api/v3/*.
organizationName	You can find your organization name when you log in to GitHub desktop and go to **Your organizations** under your profile picture dropdown.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
• ghRepository	A list of objects that map the attributes or field names of your GitHub content to Amazon

Configuration	Description
• ghCommit • ghIssueDocument • ghIssueComment • ghIssueAttachment • ghPRDocument • ghPRComment • ghPRAttachment	Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source.
isCrawlAcl	`true` to crawl the access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access and search. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
fieldForUserId	Specify the type of user ID that you want to use for ACL crawling. Specify either `email` if you want to use the user email for the user ID, or `username` if you want to use the user name for the user ID. If you don't specify an option then `email` is used by default.
repositoryFilter	A list of names of the specific repositories and branch names you want to index.
crawlRepository	`true` to crawl repositories.
crawlRepositoryDocuments	`true` to crawl repository documents.

Configuration	Description
crawlIssue	`true` to crawl issues.
crawlIssueComment	`true` to crawl issue comments.
crawlIssueCommentAttachment	`true` to crawl issue comment attachments.
crawlPullRequest	`true` to crawl pull requests.
crawlPullRequestComment	`true` to crawl pull request comments.
crawlPullRequestCommentAttachment	`true` to crawl pull request comment attachments.
• inclusionFolderNamePatterns • inclusionFileTypePatterns • inclusionFileNamePatterns	A list of regular expression patterns to include certain content in your GitHub data source. Content that matches the patterns are included in the index. Content that doesn't match the patterns are excluded from the index. If any content matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the content isn't included in the index.
• exclusionFolderNamePatterns • exclusionFileTypePatterns • exclusionFileNamePatterns	A list of regular expression patterns to exclude certain content in your GitHub data source. Content that matches the patterns are excluded from the index. Content that doesn't match the patterns are included in the index. If any content matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the content isn't included in the index.
type	The type of data source. Specify `GITHUB` as your data source type.

Configuration	Description
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

Configuration	Description
secretArn	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your GitHub. The secret must contain a JSON structure with the following keys: `{ "personalToken": " token" }`
version	The version of this template that's currently supported.

GitHub JSON schema

The following is the GitHub JSON schema:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "connectionConfiguration": {
            "type": "object",
            "properties": {
                "repositoryEndpointMetadata": {
                    "type": "object",
                    "properties": {
                        "type": {
                            "type": "string"
                        },
                        "hostUrl": {
                            "type": "string",
                            "pattern": "https://.*"
                        },
                        "organizationName": {
                            "type": "string"
                        }
                    },
```

```
                    "required": [
                        "type",
                        "hostUrl",
                        "organizationName"
                    ]
                }
            },
            "required": [
                "repositoryEndpointMetadata"
            ]
        },
        "repositoryConfigurations": {
            "type": "object",
            "properties": {
                "ghRepository": {
                    "type": "object",
                    "properties": {
                        "fieldMappings": {
                            "type": "array",
                            "items": [
                                {
                                    "type": "object",
                                    "properties": {
                                        "indexFieldName": {
                                            "type": "string"
                                        },
                                        "indexFieldType": {
                                            "type": "string",
                                            "enum": [
                                                "STRING",
                                                "STRING_LIST",
                                                "DATE"
                                            ]
                                        },
                                        "dataSourceFieldName": {
                                            "type": "string"
                                        },
                                        "dateFieldFormat": {
                                            "type": "string",
                                            "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                                        }
                                    },
                                    "required": [
                                        "indexFieldName",
```

```
                                        "indexFieldType",
                                        "dataSourceFieldName"
                                    ]
                                }
                            ]
                        }
                    },
                    "required": [
                        "fieldMappings"
                    ]
                },
                "ghCommit": {
                    "type": "object",
                    "properties": {
                        "fieldMappings": {
                            "type": "array",
                            "items": [
                                {
                                    "type": "object",
                                    "properties": {
                                        "indexFieldName": {
                                            "type": "string"
                                        },
                                        "indexFieldType": {
                                            "type": "string",
                                            "enum": [
                                                "STRING",
                                                "STRING_LIST",
                                                "DATE"
                                            ]
                                        },
                                        "dataSourceFieldName": {
                                            "type": "string"
                                        },
                                        "dateFieldFormat": {
                                            "type": "string",
                                            "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                                        }
                                    },
                                    "required": [
                                        "indexFieldName",
                                        "indexFieldType",
                                        "dataSourceFieldName"
                                    ]
```

```
                    }
                ]
            }
        },
        "required": [
            "fieldMappings"
        ]
    },
    "ghIssueDocument": {
        "type": "object",
        "properties": {
            "fieldMappings": {
                "type": "array",
                "items": [
                    {
                        "type": "object",
                        "properties": {
                            "indexFieldName": {
                                "type": "string"
                            },
                            "indexFieldType": {
                                "type": "string",
                                "enum": [
                                    "STRING",
                                    "STRING_LIST",
                                    "DATE"
                                ]
                            },
                            "dataSourceFieldName": {
                                "type": "string"
                            },
                            "dateFieldFormat": {
                                "type": "string",
                                "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                            }
                        },
                        "required": [
                            "indexFieldName",
                            "indexFieldType",
                            "dataSourceFieldName"
                        ]
                    }
                ]
            }
```

```
                    },
                    "required": [
                        "fieldMappings"
                    ]
                },
                "ghIssueComment": {
                    "type": "object",
                    "properties": {
                        "fieldMappings": {
                            "type": "array",
                            "items": [
                                {
                                    "type": "object",
                                    "properties": {
                                        "indexFieldName": {
                                            "type": "string"
                                        },
                                        "indexFieldType": {
                                            "type": "string",
                                            "enum": [
                                                "STRING",
                                                "STRING_LIST",
                                                "DATE"
                                            ]
                                        },
                                        "dataSourceFieldName": {
                                            "type": "string"
                                        },
                                        "dateFieldFormat": {
                                            "type": "string",
                                            "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                                        }
                                    },
                                    "required": [
                                        "indexFieldName",
                                        "indexFieldType",
                                        "dataSourceFieldName"
                                    ]
                                }
                            ]
                        }
                    },
                    "required": [
                        "fieldMappings"
```

```
                ]
            },
            "ghIssueAttachment": {
                "type": "object",
                "properties": {
                    "fieldMappings": {
                        "type": "array",
                        "items": [
                            {
                                "type": "object",
                                "properties": {
                                    "indexFieldName": {
                                        "type": "string"
                                    },
                                    "indexFieldType": {
                                        "type": "string",
                                        "enum": [
                                            "STRING",
                                            "STRING_LIST",
                                            "DATE"
                                        ]
                                    },
                                    "dataSourceFieldName": {
                                        "type": "string"
                                    },
                                    "dateFieldFormat": {
                                        "type": "string",
                                        "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                                    }
                                },
                                "required": [
                                    "indexFieldName",
                                    "indexFieldType",
                                    "dataSourceFieldName"
                                ]
                            }
                        ]
                    }
                },
                "required": [
                    "fieldMappings"
                ]
            },
            "ghPRDocument": {
```

```
                    "type": "object",
                    "properties": {
                        "fieldMappings": {
                            "type": "array",
                            "items": [
                                {
                                    "type": "object",
                                    "properties": {
                                        "indexFieldName": {
                                            "type": "string"
                                        },
                                        "indexFieldType": {
                                            "type": "string",
                                            "enum": [
                                                "STRING",
                                                "STRING_LIST",
                                                "DATE"
                                            ]
                                        },
                                        "dataSourceFieldName": {
                                            "type": "string"
                                        },
                                        "dateFieldFormat": {
                                            "type": "string",
                                            "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                                        }
                                    },
                                    "required": [
                                        "indexFieldName",
                                        "indexFieldType",
                                        "dataSourceFieldName"
                                    ]
                                }
                            ]
                        }
                    },
                    "required": [
                        "fieldMappings"
                    ]
                },
                "ghPRComment": {
                    "type": "object",
                    "properties": {
                        "fieldMappings": {
```

```
                        "type": "array",
                        "items": [
                            {
                                "type": "object",
                                "properties": {
                                    "indexFieldName": {
                                        "type": "string"
                                    },
                                    "indexFieldType": {
                                        "type": "string",
                                        "enum": [
                                            "STRING",
                                            "STRING_LIST",
                                            "DATE"
                                        ]
                                    },
                                    "dataSourceFieldName": {
                                        "type": "string"
                                    },
                                    "dateFieldFormat": {
                                        "type": "string",
                                        "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                                    }
                                },
                                "required": [
                                    "indexFieldName",
                                    "indexFieldType",
                                    "dataSourceFieldName"
                                ]
                            }
                        ]
                    }
                },
                "required": [
                    "fieldMappings"
                ]
            },
            "ghPRAttachment": {
                "type": "object",
                "properties": {
                    "fieldMappings": {
                        "type": "array",
                        "items": [
                            {
```

```
                                    "type": "object",
                                    "properties": {
                                        "indexFieldName": {
                                            "type": "string"
                                        },
                                        "indexFieldType": {
                                            "type": "string",
                                            "enum": [
                                                "STRING",
                                                "STRING_LIST",
                                                "DATE"
                                            ]
                                        },
                                        "dataSourceFieldName": {
                                            "type": "string"
                                        },
                                        "dateFieldFormat": {
                                            "type": "string",
                                            "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                                        }
                                    },
                                    "required": [
                                        "indexFieldName",
                                        "indexFieldType",
                                        "dataSourceFieldName"
                                    ]
                                }
                            ]
                        }
                    },
                    "required": [
                        "fieldMappings"
                    ]
                }
            }
        },
        "additionalProperties": {
            "type": "object",
            "properties": {
                "isCrawlAcl": {
                    "type": "boolean"
                },
                "fieldForUserId": {
                    "type": "string"
```

```
                 },
                 "crawlRepository": {
                     "type": "boolean"
                 },
                 "crawlRepositoryDocuments": {
                     "type": "boolean"
                 },
                 "crawlIssue": {
                     "type": "boolean"
                 },
                 "crawlIssueComment": {
                     "type": "boolean"
                 },
                 "crawlIssueCommentAttachment": {
                     "type": "boolean"
                 },
                 "crawlPullRequest": {
                     "type": "boolean"
                 },
                 "crawlPullRequestComment": {
                     "type": "boolean"
                 },
                 "crawlPullRequestCommentAttachment": {
                     "type": "boolean"
                 },
                 "repositoryFilter": {
                     "type": "array",
                     "items": [
                         {
                             "type": "object",
                             "properties": {
                                 "repositoryName": {
                                     "type": "string"
                                 },
                                 "branchNameList": {
                                     "type": "array",
                                     "items": {
                                         "type": "string"
                                     }
                                 }
                             }
                         }
                     ]
                 },
```

```
                "inclusionFolderNamePatterns": {
                    "type": "array",
                    "items": {
                        "type": "string"
                    }
                },
                "inclusionFileTypePatterns": {
                    "type": "array",
                    "items": {
                        "type": "string"
                    }
                },
                "inclusionFileNamePatterns": {
                    "type": "array",
                    "items": {
                        "type": "string"
                    }
                },
                "exclusionFolderNamePatterns": {
                    "type": "array",
                    "items": {
                        "type": "string"
                    }
                },
                "exclusionFileTypePatterns": {
                    "type": "array",
                    "items": {
                        "type": "string"
                    }
                },
                "exclusionFileNamePatterns": {
                    "type": "array",
                    "items": {
                        "type": "string"
                    }
                }
            },
            "required": []
        },
        "type": {
            "type": "string",
            "pattern": "GITHUB"
        },
        "syncMode": {
```

```
            "type": "string",
            "enum": [
                "FULL_CRAWL",
                "FORCED_FULL_CRAWL",
                "CHANGE_LOG"
            ]
        },
        "enableIdentityCrawler": {
            "type": "boolean"
        },
        "secretArn": {
            "type": "string",
            "minLength": 20,
            "maxLength": 2048
        }
    },
    "version": {
        "type": "string",
        "anyOf": [
            {
                "pattern": "1.0.0"
            }
        ]
    },
    "required": [
        "connectionConfiguration",
        "repositoryConfigurations",
        "syncMode",
        "additionalProperties",
        "enableIdentityCrawler"
    ]
}
```

Gmail template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as `GMAIL`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Gmail JSON schema.

The following table describes the parameters of the Gmail JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source. This data source does not specify an endpoint in `repositoryEndpointMetadata` . Rather, the connection information is included in an AWS Secrets Manager secret that you provide the `secretArn` .
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings. Specify the type of data source and the secret ARN.
• message • attachments	A list of objects that map the attributes or field names of your Gmail messages and attachments to Amazon Kendra index field names. For more information, see [Mapping data source fields]().
additionalProperties	Additional configuration options for your content in your data source.
• inclusionLabelNamePatterns • exclusionLabelNamePatterns • inclusionAttachmentTypePatterns • exclusionAttachmentTypePatterns • inclusionAttachmentNamePatterns • exclusionAttachmentNamePatterns • inclusionSubjectFilter • exclusionSubjectFilter • isSubjectAnd	A list of regular expression patterns to include or exclude messages with specific subject names in your Gmail data source. Files that match the patterns are included in the index. If a file matches both an inclusion and an exclusion pattern, the exclusion pattern takes precedence, and the file isn't included in the index.

Configuration	Description
• inclusionFromFilter • exclusionFromFilter • inclusionToFilter • exclusionToFilter • inclusionCcFilter • exclusionCcFilter • inclusionBccFilter • exclusionBccFilter	
beforeDateFilter	Specify messages and attachments to be included before a certain date.
afterDateFilter	Specify messages and attachments to be included after a certain date.
isCrawlAttachment	A Boolean value to choose whether you want to crawl attachments. Messages are automatic ally crawled.
type	The type of data source. Specify `GMAIL` as your data source type.
shouldCrawlDraftMessages	A Boolean value to choose whether you want to crawl draft messages.

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. **Important** Because there is no API to update permanently deleted Gmail messages, any new, modified, or deleted content sync: • Won't remove messages that were permanently deleted from Gmail from your Amazon Kendra index • Won't sync changes in Gmail email labels To sync your Gmail data source label changes and permanently deleted email messages to your Amazon Kendra index, you must run full crawls periodically.

Configuration	Description
secretARN	The Amazon Resource Name (ARN) of a Secrets Manager secret that contains the key-value pairs required to connect to your Gmail. The secret must contain a JSON structure with the following keys: `{ "adminAccountEmailId": " service account email", "clientEmailId": " user account email", "privateKey": " private key" }`
version	The version of the template that is currently supported.

Gmail JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
      }
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "message": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
```

```
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
                    "enum": ["STRING", "STRING_LIST", "DATE"]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  },
                  "dateFieldFormat": {
                    "type": "string"
                  }
                },
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        }
      },
      "attachments": {
        "type": "object",
        "properties": {
          "fieldMappings": {
            "type": "array",
            "items": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
                    "enum": ["STRING"]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
```

```
                  }
                },
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        }
      }
    },
    "required": []
  },
  "additionalProperties": {
    "type": "object",
    "properties": {
      "inclusionLabelNamePatterns": {
        "type": "array",
        "items": {
          "type": "string"
        }
      },
      "exclusionLabelNamePatterns": {
        "type": "array",
        "items": {
          "type": "string"
        }
      },
      "inclusionAttachmentTypePatterns": {
        "type": "array",
        "items": {
          "type": "string"
        }
      },
      "exclusionAttachmentTypePatterns": {
        "type": "array",
        "items": {
          "type": "string"
        }
      },
      "inclusionAttachmentNamePatterns": {
        "type": "array",
```

```
          "items": {
            "type": "string"
          }
        },
        "exclusionAttachmentNamePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionSubjectFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionSubjectFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "isSubjectAnd": {
          "type": "boolean"
        },
        "inclusionFromFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionFromFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionToFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionToFilter": {
```

```
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionCcFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionCcFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionBccFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionBccFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "beforeDateFilter": {
          "anyOf": [
            {
              "type": "string",
              "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}Z$"
            },
            {
              "type": "string",
              "pattern": ""
            }
          ]
        },
        "afterDateFilter": {
          "anyOf": [
            {
```

```
              "type": "string",
              "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}Z$"
            },
            {
              "type": "string",
              "pattern": ""
            }
          ]
        },
        "isCrawlAttachment": {
          "type": "boolean"
        },
        "shouldCrawlDraftMessages": {
          "type": "boolean"
        }
      },
      "required": [
        "isCrawlAttachment",
        "shouldCrawlDraftMessages"
      ]
    },
    "type" : {
      "type" : "string",
      "pattern": "GMAIL"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL"
      ]
    },
    "secretArn": {
      "type": "string"
    },
    "version": {
      "type": "string",
      "anyOf": [
        {
          "pattern": "1.0.0"
        }
      ]
    }
  },
```

```
  "required": [
    "connectionConfiguration",
    "repositoryConfigurations",
    "additionalProperties",
    "syncMode",
    "secretArn",
    "type"
  ]
}
```

Google Drive template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as `GOOGLEDRIVE2`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Google Drive JSON schema.

The following table describes the parameters of the Google Drive JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the data source.
repositoryEndpointMetadata	The endpoint information for the data source. This data source does not specify an endpoint. You choose your authentication type: `serviceAccount` and `OAuth2`. The connection information is included in an AWS Secrets Manager secret that you provide the `secretArn` .
authType	Choose between `serviceAccount` and `OAuth2` based on your use case.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.

Configuration	Description
• file • comment	A list of objects that map the attributes or field names of your Google Drive to Amazon Kendra index field names. For more informati on, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source
• maxFileSizeInMegaBytes	Specify a file size limit in MBs that Amazon Kendra should crawl.
• iscrawlComment	`true` to crawl comments in your Google Drive data source.
• isCrawlMyDriveAndSharedWithMe	`true` to crawl MyDrive and Shared With Me Drives in your Google Drive data source.
• isCrawlSharedDrives	`true` to crawl Shared Drives in your Google Drive data source.
isCrawlAcl	`true` to crawl the access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access and search. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

Configuration	Description
• excludeUserAccounts • excludeSharedDrives • excludeMimeTypes • exclusionFileTypePatterns • exclusionFileNamePatterns • exclusionFilePathFilter	A list of regular expression patterns to *exclude* certain files in your Google Drive data source. Files that match the patterns are excluded from the index. Files that don't match the patterns are included in the index. If a file matches both an exclusion and inclusion pattern, the exclusion pattern takes precedenc e, and the file isn't included in the index.
• includeUserAccounts • includeSharedDrives • includeMimeTypes • inclusionFileTypePatterns • inclusionFileNamePatterns • inclusionFilePathFilter	A list of regular expression patterns to *include* certain files in your Google Drive data source. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the file isn't included in the index.
type	The type of data source. Specify `GOOOGLEDRIVEV2` as your data source type.
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • FORCED_FULL_CRAWL to freshly index all content, replacing existing content each time your data source syncs with your index. • FULL_CRAWL to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • CHANGE_LOG to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

Configuration	Description

secretARN	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Google Drive. The secret must contain a JSON structure with the following keys:
	
	If using Google Service Account authentication:
	
	`{ "clientEmail": " user account email", "adminAccountEmail": " service account email", "privateKey": " private key" }`
	
	If using OAuth 2.0 authentication:
	
	`{ "clientID": " OAuth client ID", "clientSecret": " client secret", "refreshToken": " refresh token" }`

version	The version of this template that is currently supported.

Google Drive JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
```

```
          "repositoryEndpointMetadata": {
            "type": "object",
            "properties": {
              "authType": {
                "type": "string",
                "enum": [
                  "serviceAccount",
                  "OAuth2"
                ]
              }
            },
            "required": [
              "authType"
            ]
          }
        },
        "required": [
          "repositoryEndpointMetadata"
        ]
      },
      "repositoryConfigurations": {
        "type": "object",
        "properties": {
          "file": {
            "type": "object",
            "properties": {
              "fieldMappings": {
                "type": "array",
                "items": [
                  {
                    "type": "object",
                    "properties": {
                      "indexFieldName": {
                        "type": "string"
                      },
                      "indexFieldType": {
                        "type": "string",
                        "enum": [
                          "STRING",
                          "DATE",
                          "STRING_LIST",
                          "LONG"
                        ]
                      },
```

```
                "dataSourceFieldName": {
                  "type": "string"
                },
                "dateFieldFormat": {
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                }
              },
              "required": [
                "indexFieldName",
                "indexFieldType",
                "dataSourceFieldName"
              ]
            }
          ]
        }
      },
      "required": [
        "fieldMappings"
      ]
    },
    "comment": {
      "type": "object",
      "properties": {
        "fieldMappings": {
          "type": "array",
          "items": [
            {
              "type": "object",
              "properties": {
                "indexFieldName": {
                  "type": "string"
                },
                "indexFieldType": {
                  "type": "string",
                  "enum": [
                    "STRING",
                    "DATE",
                    "STRING_LIST"
                  ]
                },
                "dataSourceFieldName": {
                  "type": "string"
                },
```

```
                "dateFieldFormat": {
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                }
              },
              "required": [
                "indexFieldName",
                "indexFieldType",
                "dataSourceFieldName"
              ]
            }
          ]
        }
      },
      "required": [
        "fieldMappings"
      ]
    }
  }
},
"additionalProperties": {
  "type": "object",
  "properties": {
    "maxFileSizeInMegaBytes": {
      "type": "string"
    },
    "isCrawlComment": {
      "type": "boolean"
    },
    "isCrawlMyDriveAndSharedWithMe": {
      "type": "boolean"
    },
    "isCrawlSharedDrives": {
      "type": "boolean"
    },
    "isCrawlAcl": {
      "type": "boolean"
    },
    "excludeUserAccounts": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
```

```
        "excludeSharedDrives": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "excludeMimeTypes": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "includeUserAccounts": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "includeSharedDrives": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "includeMimeTypes": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "includeTargetAudienceGroup": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionFileTypePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionFileNamePatterns": {
          "type": "array",
```

```
          "items": {
            "type": "string"
          }
        },
        "exclusionFileTypePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionFileNamePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionFilePathFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionFilePathFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      }
    },
    "type": {
      "type": "string",
      "pattern": "GOOGLEDRIVEV2"
    },
    "enableIdentityCrawler": {
      "type": "boolean"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL",
        "CHANGE_LOG"
      ]
```

```
    },
    "secretArn": {
      "type": "string",
      "minLength": 20,
      "maxLength": 2048
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
    "connectionConfiguration",
    "repositoryConfigurations",
    "syncMode",
    "additionalProperties",
    "secretArn",
    "type"
  ]
}
```

IBM DB2 template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as JDBC, the database type as db2, a secret for your authentication credentials, and other necessary configurations. You then specify TEMPLATE as the Type when you call CreateDataSource.

You can use the template provided in this developer guide. See IBM DB2 JSON schema.

The following table describes the parameters of the IBM DB2 JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.

Configuration	Description
repositoryEndpointMetadata	Required configuration information for connecting your data source. • dbType—The type of Java database that you use, whether `mysql`, `db2`, `postgresql` , `oracle`, or `sqlserver` . • dbHost—The database host name. • dbPort—The database port. • dbInstance—The database instance.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings. Specify the type of data source and the secret ARN.
document	A list of objects that map the attributes or field names of your database content to Amazon Kendra index field names. For more information, see [Mapping data source fields]().
additionalProperties	Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.
primaryKey	Provide the primary key for the database table. This identifies a table within your database.
titleColumn	Provide the name of the document title column within your database table.
bodyColumn	Provide the name of the document title column within your database table.

Configuration	Description
sqlQuery	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
timestampColumn	Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
timestampFormat	Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.
timezone	Enter the name of the column which contains time zones for the content to be crawled.
changeDetectingColumns	Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns
allowedUsersColumns	Enter the name of the column which contains User IDs to be allowed access to content.
allowedGroupsColumn	Enter the name of the column which contains User IDs to be allowed access to content.
sourceURIColumn	Enter the name of the column which contains Source URLs to be indexed.
isSslEnabled	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.

<table>
<tr><th>Configuration</th><th>Description</th></tr>
<tr><td>type</td><td>The type of data source. Specify <code>JDBC</code> as your data source type.</td></tr>
<tr><td>syncMode</td><td>Specify how Amazon Kendra should update your index when your data source content changes. You can choose between:
<code>FORCED_FULL_CRAWL</code> to freshly index all content, replacing existing content each time your data source syncs with your index.
<code>FULL_CRAWL</code> to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
<code>CHANGE_LOG</code> to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.</td></tr>
<tr><td>secretArn</td><td>The Amazon Resource Name (ARN) of a Secrets Manager secret that contains user name and password required to connect to your database. The secret must contain a JSON structure with the following keys:<pre>{
    "user name": "database user name",
    "password": " password"
}</pre></td></tr>
</table>

Configuration	Description
version	The version of the template that is currently supported.

IBM DB2 JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "dbType": {
              "type": "string",
              "enum": [
                "mysql",
                "db2",
                "postgresql",
                "oracle",
                "sqlserver"
              ]
            },
            "dbHost": {
              "type": "string"
            },
            "dbPort": {
              "type": "string"
            },
            "dbInstance": {
              "type": "string"
            }
          },
          "required": [
            "dbType",
            "dbHost",
            "dbPort",
            "dbInstance"
```

```
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "document": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string"
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
```

```
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "primaryKey": {
          "type": "string"
        },
        "titleColumn": {
          "type": "string"
        },
        "bodyColumn": {
          "type": "string"
        },
        "sqlQuery": {
          "type": "string",
          "not": {
            "pattern": ";+"
          }
        },
        "timestampColumn": {
          "type": "string"
        },
        "timestampFormat": {
          "type": "string"
        },
        "timezone": {
          "type": "string"
        },
        "changeDetectingColumns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "allowedUsersColumn": {
          "type": "string"
        },
        "allowedGroupsColumn": {
          "type": "string"
        },
        "sourceURIColumn": {
          "type": "string"
        },
```

```
        "isSslEnabled": {
          "type": "boolean"
        }
      },
      "required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
    },
    "type" : {
      "type" : "string",
      "pattern": "JDBC"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL",
        "CHANGE_LOG"
      ]
    },
    "secretArn": {
      "type": "string"
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
      "connectionConfiguration",
      "repositoryConfigurations",
      "syncMode",
      "additionalProperties",
      "secretArn",
      "type"
  ]
}
```

Microsoft Exchange template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the tenant ID as as a part of the connection configuration or repository endpoint details. Also specify the type of data source as `MSEXCHANGE`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Microsoft Exchange JSON schema.

The following table describes the parameters of the Microsoft Exchange JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
tenantId	The Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal or in your OAuth application.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
• email • attachment • calendar • contacts • notes	A list of objects that map the attributes or field names of your Microsoft Exchange data source to Amazon Kendra index fields. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for content in your data source
inclusionPatterns	A list of regular expression patterns to *include* certain files in your Microsoft Exchange data

Configuration	Description
	source. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedenc e and the file isn't included in the index.
exclusionPatterns	A list of regular expression patterns to *exclude* certain files in your Microsoft Exchange data source. Files that match the patterns are excluded from the index. Files that don't match the patterns are included in the index. If a file matches both an exclusion and inclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.
• inclusionUsersList • inclusionUsersFileName • inclusionDomainUsers	A list of regular expression patterns to *include* certain users and user files in your Microsofo t Exchange data source. Users that match the patterns are included in the index. Users that don't match the patterns are excluded from the index. If a user matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence and the user isn't included in the index.

Configuration	Description
• exclusionUsersList • exclusionUsersFileName • exclusionDomainUsers	A list of regular expression patterns to *exclude* certain users and user files in your Microsoft Exchange data source. Users that match the patterns are excluded from the index. Users that don't match the patterns are included in the index. If a user matches both an exclusion and inclusion pattern, the exclusion pattern takes precedence and the user isn't included in the index.
s3bucketName	The name of your S3 bucket if that you want to use.
• crawlCalendar • crawlNotes • crawlContacts • crawlFolderAcl	`true` to crawl these types of content and access control information your Microsoft Exchange data source.
startCalendarDateTime	You can configure a specific start date-time for your calendar content.
endCalendarDateTime	You can configure a specific end date-time for calendar content.
subject	You can configure a specific subject line for your mail content.
emailFrom	You can configure a specific email for your 'From' or sender mail content.
emailTo	You can configure a specific email for your 'To' or recipient mail content.

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
type	The type of data source. Specify `MSEXCHANGE` as your data source type.
secretARN	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Microsoft Exchange. This includes your client ID and your client secret that is generated when you create an OAuth application in the Azure portal.
version	The version of this template that is currently supported.

Microsoft Exchange JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "tenantId": {
              "type": "string",
              "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]
{12}$",
              "minLength": 36,
              "maxLength": 36
            }
          },
          "required": ["tenantId"]
        }
      }
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "email": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string",
                      "enum": ["STRING", "STRING_LIST", "DATE"]
                    },
                    "dataSourceFieldName": {
```

```
                "type": "string"
              },
              "dateFieldFormat": {
                "type": "string",
                "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
              }
            },
            "required": [
              "indexFieldName",
              "indexFieldType",
              "dataSourceFieldName"
            ]
          }
        ]
      }
    },
    "required": [
      "fieldMappings"
    ]
  },
  "attachment": {
    "type": "object",
    "properties": {
      "fieldMappings": {
        "type": "array",
        "items": [
          {
            "type": "object",
            "properties": {
              "indexFieldName": {
                "type": "string"
              },
              "indexFieldType": {
                "type": "string",
                "enum": ["STRING", "DATE","LONG"]
              },
              "dataSourceFieldName": {
                "type": "string"
              },
              "dateFieldFormat": {
                "type": "string",
                "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
              }
            },
```

```
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        },
        "required": [
          "fieldMappings"
        ]
      },
      "calendar": {
        "type": "object",
        "properties": {
          "fieldMappings": {
            "type": "array",
            "items": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
                    "enum": ["STRING", "STRING_LIST", "DATE"]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  },
                  "dateFieldFormat": {
                    "type": "string",
                    "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                  }
                },
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
```

```
            }
          },
          "required": [
            "fieldMappings"
          ]
        },
        "contacts": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string",
                      "enum": ["STRING", "STRING_LIST", "DATE"]
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        },
        "notes": {
```

```
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string",
                      "enum": ["STRING", "DATE"]
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": ["email"
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "inclusionPatterns": {
```

```
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "exclusionPatterns": {
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "inclusionUsersList": {
            "type": "array",
            "items": {
              "type": "string",
              "format": "email"
            }
          },
          "exclusionUsersList": {
            "type": "array",
            "items": {
              "type": "string",
              "format": "email"
            }
          },
          "s3bucketName": {
            "type": "string"
          },
          "inclusionUsersFileName": {
            "type": "string"
          },
          "exclusionUsersFileName": {
            "type": "string"
          },
          "inclusionDomainUsers": {
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "exclusionDomainUsers": {
            "type": "array",
            "items": {
              "type": "string"
```

```
          }
        },
        "crawlCalendar": {
          "type": "boolean"
        },
        "crawlNotes": {
          "type": "boolean"
        },
        "crawlContacts": {
          "type": "boolean"
        },
        "crawlFolderAcl": {
          "type": "boolean"
        },
        "startCalendarDateTime": {
          "anyOf": [
            {
              "type": "string",
              "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}Z$"
            },
            {
              "type": "string",
              "pattern": ""
            }
          ]
        },
        "endCalendarDateTime": {
          "anyOf": [
            {
            "type": "string",
            "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}Z$"
            },
            {
              "type": "string",
              "pattern": ""
            }
          ]
        },
        "subject": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
```

```
        "emailFrom": {
          "type": "array",
          "items": {
            "type": "string",
            "format": "email"
          }
        },
        "emailTo": {
          "type": "array",
          "items": {
            "type": "string",
            "format": "email"
          }
        }
      },
      "required": [
      ]
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL",
        "CHANGE_LOG"
      ]
    },
    "type" : {
      "type" : "string",
      "pattern": "MSEXCHANGE"
    },
    "secretArn": {
      "type": "string"
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
    "connectionConfiguration",
```

```
    "repositoryConfigurations",
    "syncMode",
    "additionalProperties",
    "secretArn",
    "type"
  ]
}
```

Microsoft OneDrive template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the tenant ID as part of the connection configuration or repository endpoint details. Also specify the type of data source as `ONEDRIVEV2`, and a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Microsoft OneDrive JSON schema.

The following table describes the parameters of the Microsoft OneDrive JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
tenantId	The Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal or in your OAuth application.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
file	A list of objects that map the attributes or field names of your Microsoft OneDrive files to Amazon Kendra index field names. For more information, see Mapping data source fields.

Configuration	Description
additionalProperties	Additional configuration options for your content in your data source
• userNameFilter • userFilterPath • inclusionFileTypePatterns • exclusionFileTypePatterns • inclusionFileNamePatterns • exclusionFileNamePatterns • inclusionFilePathPatterns • exclusionFilePathPatterns • inclusionOneNoteSectionNamePatterns • exclusionOneNoteSectionNamePatterns • inclusionOneNotePageNamePatterns • exclusionOneNotepageNamePatterns	You can choose to index specific files, OneNote sections, OneNote pages, and filter by user name.
isUserNameOnS3	`true` to provide a list of user names in a file stored in an Amazon S3.
type	The type of data source. Specify `ONEDRIVEV2` as your data source type.
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.

<table>
<tr><th>Configuration</th><th>Description</th></tr>
<tr><td>type</td><td>The type of data source. Specify <code>ONEDRIVEV2</code> as your data source type.</td></tr>
<tr><td>syncMode</td><td>Specify how Amazon Kendra should update your index when your data source content changes. You can choose between:
<code>FORCED_FULL_CRAWL</code> to freshly index all content, replacing existing content each time your data source syncs with your index.
<code>FULL_CRAWL</code> to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
<code>CHANGE_LOG</code> to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.</td></tr>
<tr><td>secretARN</td><td>The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Microsoft OneDrive. The secret must contain a JSON structure with the following keys:<pre>{
    "clientId": " client ID",
    "clientSecret": " client secret"
}</pre></td></tr>
</table>

Configuration	Description
version	The version of this template that is currently supported.

Microsoft OneDrive JSON schema

```
{
 "$schema": "http://json-schema.org/draft-04/schema#",
 "type": "object",
 "properties": {
  "connectionConfiguration": {
   "type": "object",
   "properties": {
    "repositoryEndpointMetadata": {
     "type": "object",
     "properties": {
      "tenantId": {
       "type": "string",
       "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$",
       "minLength": 36,
       "maxLength": 36
      }
     },
     "required": [
      "tenantId"
     ]
    }
   },
   "required": [
    "repositoryEndpointMetadata"
   ]
  },
  "repositoryConfigurations": {
   "type": "object",
   "properties": {
    "file": {
     "type": "object",
     "properties": {
      "fieldMappings": {
       "type": "array",
       "items": [
```

```
        {
         "type": "object",
         "properties": {
          "indexFieldName": {
           "type": "string"
          },
          "indexFieldType": {
           "type": "string",
           "enum": [
            "STRING",
            "STRING_LIST",
            "DATE",
            "LONG"
           ]
          },
          "dataSourceFieldName": {
           "type": "string"
          },
          "dateFieldFormat": {
           "type": "string",
           "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
          }
         },
         "required": [
          "indexFieldName",
          "indexFieldType",
          "dataSourceFieldName"
         ]
        }
       ]
      }
     },
     "required": [
      "fieldMappings"
     ]
    }
   }
  },
  "additionalProperties": {
   "type": "object",
   "properties": {
    "userNameFilter": {
     "type": "array",
     "items": {
```

```
        "type": "string"
       }
      },
      "userFilterPath": {
       "type": "string"
      },
      "isUserNameOnS3": {
       "type": "boolean"
      },
      "inclusionFileTypePatterns": {
       "type": "array",
       "items": {
        "type": "string"
       }
      },
      "exclusionFileTypePatterns": {
       "type": "array",
       "items": {
        "type": "string"
       }
      },
      "inclusionFileNamePatterns": {
       "type": "array",
       "items": {
        "type": "string"
       }
      },
      "exclusionFileNamePatterns": {
       "type": "array",
       "items": {
        "type": "string"
       }
      },
      "inclusionFilePathPatterns": {
       "type": "array",
       "items": {
        "type": "string"
       }
      },
      "exclusionFilePathPatterns": {
       "type": "array",
       "items": {
        "type": "string"
       }
```

```
    },
    "inclusionOneNoteSectionNamePatterns": {
     "type": "array",
     "items": {
      "type": "string"
     }
    },
    "exclusionOneNoteSectionNamePatterns": {
     "type": "array",
     "items": {
      "type": "string"
     }
    },
    "inclusionOneNotePageNamePatterns": {
     "type": "array",
     "items": {
      "type": "string"
     }
    },
    "exclusionOneNotePageNamePatterns": {
     "type": "array",
     "items": {
      "type": "string"
     }
    }
   },
   "required": []
  },

  "enableIdentityCrawler": {
   "type": "boolean"
  },
  "type": {
   "type": "string",
   "pattern": "ONEDRIVEV2"
  },
  "syncMode": {
   "type": "string",
   "enum": [
    "FULL_CRAWL",
    "FORCED_FULL_CRAWL",
    "CHANGE_LOG"
   ]
  },
```

```
   "secretArn": {
    "type": "string",
    "minLength": 20,
    "maxLength": 2048
   }
  },
  "version": {
   "type": "string",
   "anyOf": [
    {
     "pattern": "1.0.0"
    }
   ]
  },
  "required": [
   "connectionConfiguration",
   "repositoryConfigurations",
   "syncMode",
   "additionalProperties",
   "secretArn",
   "type"
  ]
}
```

Microsoft SharePoint template schema

You include a JSON that contains the data source schema as part of TemplateConfiguration object. You provide the SharePoint site URL/URLs, domain, and also a tenant ID if required as a part of the connection configuration or repository endpoint details. Also specify the type of data source as `SHAREPOINTV2`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the **Type** when you call CreateDataSource.

You can use the template provided in this developer guide. See SharePoint JSON schema.

The following table describes the parameters of the Microsoft SharePoint JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source
repositoryEndpointMetadata	The endpoint information for the data source

Configuration	Description
tenantId	The tenant id of your SharePoint account.
domain	The domain of your SharePoint account.
siteUrls	The host URLs of your SharePoint account.
repositoryAdditionalProperties	Additional properties to connect with the repository/data source endpoint.
s3bucketName	The name of the Amazon S3 bucket that stores your Azure AD self-signed X.509 certificate.
s3certificateName	The name of the Azure AD self-signed X.509 certificate stored in your Amazon S3 bucket.
authType	The type of authentication that you use, whether `OAuth2`, `OAuth2Certificate`, `OAuth2App`, `Basic`, `OAuth2_RefreshToken`, `NTLM`, or `Kerberos`.
version	The SharePoint version that you use, whether `Server` or `Online`.
onPremVersion	The SharePoint Server version that you use, whether `2013`, `2016` `2019`, or `SubscriptionEdition`.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.

Configuration	Description
• event • page • file • link • attachment • comment	A list of objects that map the attributes or field names of your SharePoint content to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source.
• eventTitleFilterRegEx • pageTitleFilterRegEx • linkTitleFilterRegEx • inclusionFilePath • exclusionFilePath • inclusionFileTypePatterns • exclusionFileTypePatterns • inclusionFileNamePatterns • exclusionFileNamePatterns • inclusionOneNoteSectionNamePatterns • exclusionOneNoteSectionNamePatterns • inclusionOneNotePageNamePatterns • exclusionOneNotePageNamePatterns	A list of regular expression patterns to include/exclude certain content in your SharePoint data source. Content itmes that match the inclusion patterns are included in the index. Content items that don't match the inclusion patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the file isn't included in the index.
• crawlFiles • crawlPages • crawlEvents • crawlComments • crawlLinks • crawlAttachments	`true` to crawl these types of content.

Configuration	Description
crawlAcl	`true` to crawl the access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access and search. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
fieldForUserId	Specify either `email` if you want to use the user email for the user ID, or `userPrincipalName` if you want to use a user name for the user ID. If you don't specify an option then `email` is used by default.
aclConfiguration	Specify either `ACLWithLDAPEmailFmt` , `ACLWithManualEmailFmt` , or `ACLWithUsernameFmtM` .
emailDomain	The domain of the email. For example, "*amazon.com*".
• isCrawlLocalGroupMapping • isCrawlAdGroupMapping	`true` to crawl group mapping information.
proxyHost	The host name of the web proxy that you use, without the http:// or https:// protocol.
proxyPort	The port number used by the host URL transport protocol. Must be a numeric value between 0 and 65535.
type	Specify `SHAREPOINTV2` as your data source type

Configuration	Description
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

Configuration	Description
secretARN	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your SharePoint. For information on these key-value pairs, see Connection instructions for SharePoint Online and SharePoint Server.
version	The version of this template that is currently supported.

SharePoint JSON schema

```
{
 "$schema": "http://json-schema.org/draft-04/schema#",
 "type": "object",
 "properties": {
  "connectionConfiguration": {
   "type": "object",
   "properties": {
    "repositoryEndpointMetadata": {
     "type": "object",
     "properties": {
      "tenantId": {
       "type": "string",
       "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$",
       "minLength": 36,
       "maxLength": 36
      },
      "domain": {
       "type": "string"
      },
      "siteUrls": {
       "type": "array",
       "items": {
        "type": "string",
        "pattern": "https://.*"
       }
      },
      "repositoryAdditionalProperties": {
```

```
        "type": "object",
        "properties": {
         "s3bucketName": {
          "type": "string"
         },
         "s3certificateName": {
          "type": "string"
         },
         "authType": {
          "type": "string",
          "enum": [
           "OAuth2",
           "OAuth2Certificate",
           "OAuth2App",
           "Basic",
           "OAuth2_RefreshToken",
           "NTLM",
           "Kerberos"
          ]
         },
         "version": {
          "type": "string",
          "enum": [
           "Server",
           "Online"
          ]
         },
         "onPremVersion": {
          "type": "string",
          "enum": [
           "",
           "2013",
           "2016",
           "2019",
           "SubscriptionEdition"
          ]
         }
        },
        "required": [
         "authType",
         "version"
        ]
       }
      },
```

```
      "required": [
       "siteUrls",
       "domain",
       "repositoryAdditionalProperties"
      ]
     }
    },
    "required": [
     "repositoryEndpointMetadata"
    ]
   },
   "repositoryConfigurations": {
    "type": "object",
    "properties": {
     "event": {
      "type": "object",
      "properties": {
       "fieldMappings": {
        "type": "array",
        "items": [
         {
          "type": "object",
          "properties": {
           "indexFieldName": {
            "type": "string"
           },
           "indexFieldType": {
            "type": "string",
            "enum": [
             "STRING",
             "STRING_LIST",
             "DATE"
            ]
           },
           "dataSourceFieldName": {
            "type": "string"
           },
           "dateFieldFormat": {
            "type": "string",
            "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
           }
          },
          "required": [
           "indexFieldName",
```

```
          "indexFieldType",
          "dataSourceFieldName"
         ]
        }
       ]
      }
     },
     "required": [
      "fieldMappings"
     ]
    },
    "page": {
     "type": "object",
     "properties": {
      "fieldMappings": {
       "type": "array",
       "items": [
        {
         "type": "object",
         "properties": {
          "indexFieldName": {
           "type": "string"
          },
          "indexFieldType": {
           "type": "string",
           "enum": [
            "STRING",
            "DATE",
            "LONG"
           ]
          },
          "dataSourceFieldName": {
           "type": "string"
          },
          "dateFieldFormat": {
           "type": "string",
           "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
          }
         },
         "required": [
          "indexFieldName",
          "indexFieldType",
          "dataSourceFieldName"
         ]
```

```
          }
         ]
        }
       },
       "required": [
        "fieldMappings"
       ]
      },
      "file": {
       "type": "object",
       "properties": {
        "fieldMappings": {
         "type": "array",
         "items": [
          {
           "type": "object",
           "properties": {
            "indexFieldName": {
             "type": "string"
            },
            "indexFieldType": {
             "type": "string",
             "enum": [
              "STRING",
              "DATE",
              "LONG"
             ]
            },
            "dataSourceFieldName": {
             "type": "string"
            },
            "dateFieldFormat": {
             "type": "string",
             "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
            }
           },
           "required": [
            "indexFieldName",
            "indexFieldType",
            "dataSourceFieldName"
           ]
          }
         ]
        }
```

```
      },
      "required": [
       "fieldMappings"
      ]
     },
     "link": {
      "type": "object",
      "properties": {
       "fieldMappings": {
        "type": "array",
        "items": [
         {
          "type": "object",
          "properties": {
           "indexFieldName": {
            "type": "string"
           },
           "indexFieldType": {
            "type": "string",
            "enum": [
             "STRING",
             "STRING_LIST",
             "DATE"
            ]
           },
           "dataSourceFieldName": {
            "type": "string"
           },
           "dateFieldFormat": {
            "type": "string",
            "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
           }
          },
          "required": [
           "indexFieldName",
           "indexFieldType",
           "dataSourceFieldName"
          ]
         }
        ]
       }
      },
      "required": [
       "fieldMappings"
```

```
      ]
     },
     "attachment": {
      "type": "object",
      "properties": {
       "fieldMappings": {
        "type": "array",
        "items": [
         {
          "type": "object",
          "properties": {
           "indexFieldName": {
            "type": "string"
           },
           "indexFieldType": {
            "type": "string",
            "enum": [
             "STRING",
             "STRING_LIST",
             "DATE"
            ]
           },
           "dataSourceFieldName": {
            "type": "string"
           },
           "dateFieldFormat": {
            "type": "string",
            "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
           }
          },
          "required": [
           "indexFieldName",
           "indexFieldType",
           "dataSourceFieldName"
          ]
         }
        ]
       }
      },
      "required": [
       "fieldMappings"
      ]
     },
     "comment": {
```

```
      "type": "object",
      "properties": {
       "fieldMappings": {
        "type": "array",
        "items": [
         {
          "type": "object",
          "properties": {
           "indexFieldName": {
            "type": "string"
           },
           "indexFieldType": {
            "type": "string",
            "enum": [
             "STRING",
             "STRING_LIST",
             "DATE"
            ]
           },
           "dataSourceFieldName": {
            "type": "string"
           },
           "dateFieldFormat": {
            "type": "string",
            "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
           }
          },
          "required": [
           "indexFieldName",
           "indexFieldType",
           "dataSourceFieldName"
          ]
         }
        ]
       }
      },
      "required": [
       "fieldMappings"
      ]
     }
    }
   },
   "additionalProperties": {
    "type": "object",
```

```
    "properties": {
     "eventTitleFilterRegEx": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "pageTitleFilterRegEx": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "linkTitleFilterRegEx": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "inclusionFilePath": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "exclusionFilePath": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "inclusionFileTypePatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "exclusionFileTypePatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "inclusionFileNamePatterns": {
```

```
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "exclusionFileNamePatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "inclusionOneNoteSectionNamePatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "exclusionOneNoteSectionNamePatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "inclusionOneNotePageNamePatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "exclusionOneNotePageNamePatterns": {
      "type": "array",
      "items": {
       "type": "string"
      }
     },
     "crawlFiles": {
      "type": "boolean"
     },
     "crawlPages": {
      "type": "boolean"
     },
     "crawlEvents": {
      "type": "boolean"
     },
```

```
     "crawlComments": {
      "type": "boolean"
     },
     "crawlLinks": {
      "type": "boolean"
     },
     "crawlAttachments": {
      "type": "boolean"
     },
     "crawlListData": {
      "type": "boolean"
     },
     "crawlAcl": {
      "type": "boolean"
     },
     "fieldForUserId": {
      "type": "string"
     },
     "aclConfiguration": {
      "type": "string",
      "enum": [
       "ACLWithLDAPEmailFmt",
       "ACLWithManualEmailFmt",
       "ACLWithUsernameFmt"
      ]
     },
     "emailDomain": {
      "type": "string"
     },
     "isCrawlLocalGroupMapping": {
      "type": "boolean"
     },
     "isCrawlAdGroupMapping": {
      "type": "boolean"
     },
     "proxyHost": {
      "type": "string"
     },
     "proxyPort": {
      "type": "string"
     }
    },
    "required": [
    ]
```

```
  },
  "type": {
   "type": "string",
   "pattern": "SHAREPOINTV2"
  },
  "enableIdentityCrawler": {
   "type": "boolean"
  },
  "syncMode": {
   "type": "string",
   "enum": [
    "FULL_CRAWL",
    "FORCED_FULL_CRAWL",
    "CHANGE_LOG"
   ]
  },
  "secretArn": {
   "type": "string",
   "minLength": 20,
   "maxLength": 2048
  }
 },
 "version": {
  "type": "string",
  "anyOf": [
   {
    "pattern": "1.0.0"
   }
  ]
 },
 "required": [
  "connectionConfiguration",
  "repositoryConfigurations",
  "enableIdentityCrawler",
  "syncMode",
  "additionalProperties",
  "secretArn",
  "type"
 ]
}
```

Microsoft SQL Server template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as `JDBC`, the database type as `sqlserver`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Microsoft SQL Server JSON schema.

The following table describes the parameters of the Micorosft SQL Server JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	Required configuration information for connecting your data source. • dbType—The type of Java database that you use, whether `mysql`, `db2`, `postgresql`, `oracle`, or `sqlserver`. • dbHost—The database host name. • dbPort—The database port. • dbInstance—The database instance.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings. Specify the type of data source and the secret ARN.
document	A list of objects that map the attributes or field names of your database content to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source. Use to include or

Configuration	Description
	exclude specific content in your database data source.
primaryKey	Provide the primary key for the database table. This identifies a table within your database.
titleColumn	Provide the name of the document title column within your database table.
bodyColumn	Provide the name of the document title column within your database table.
sqlQuery	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
timestampColumn	Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
timestampFormat	Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.
timezone	Enter the name of the column which contains time zones for the content to be crawled.
changeDetectingColumns	Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns
allowedUsersColumns	Enter the name of the column which contains User IDs to be allowed access to content.

Configuration	Description
allowedGroupsColumn	Enter the name of the column which contains User IDs to be allowed access to content.
sourceURIColumn	Enter the name of the column which contains Source URLs to be indexed.
isSslEnabled	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
type	The type of data source. Specify `JDBC` as your data source type.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

Configuration	Description
secretArn	The Amazon Resource Name (ARN) of a Secrets Manager secret that contains user name and password required to connect to your database. The secret must contain a JSON structure with the following keys: `{ "user name": "database user name", "password": " password" }`
version	The version of the template that is currently supported.

Microsoft SQL Server JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "dbType": {
              "type": "string",
              "enum": [
                "mysql",
                "db2",
                "postgresql",
                "oracle",
                "sqlserver"
              ]
            },
            "dbHost": {
              "type": "string"
```

```
            },
            "dbPort": {
              "type": "string"
            },
            "dbInstance": {
              "type": "string"
            }
          },
          "required": [
            "dbType",
            "dbHost",
            "dbPort",
            "dbInstance"
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "document": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string"
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    }
                  },
                  "required": [
                    "indexFieldName",
```

```
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "primaryKey": {
          "type": "string"
        },
        "titleColumn": {
          "type": "string"
        },
        "bodyColumn": {
          "type": "string"
        },
        "sqlQuery": {
          "type": "string",
          "not": {
            "pattern": ";+"
          }
        },
        "timestampColumn": {
          "type": "string"
        },
        "timestampFormat": {
          "type": "string"
        },
        "timezone": {
          "type": "string"
        },
        "changeDetectingColumns": {
          "type": "array",
```

```
          "items": {
            "type": "string"
          }
        },
        "allowedUsersColumn": {
          "type": "string"
        },
        "allowedGroupsColumn": {
          "type": "string"
        },
        "sourceURIColumn": {
          "type": "string"
        },
        "isSslEnabled": {
          "type": "boolean"
        }
      },
      "required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
    },
    "type" : {
      "type" : "string",
      "pattern": "JDBC"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL",
        "CHANGE_LOG"
      ]
    },
    "secretArn": {
      "type": "string"
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
```

```
        "connectionConfiguration",
        "repositoryConfigurations",
        "syncMode",
        "additionalProperties",
        "secretArn",
        "type"
    ]
}
```

Microsoft Teams template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the tenant ID as a part of the connection configuration or repository endpoint details. Also specify the type of data source as `MSTEAMS`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Microsoft Teams JSON schema.

The following table describes the parameters of the Microsoft Teams JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
tenantId	The Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal or in your OAuth application.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
• chatMessage • chatAttachment • channelPost	A list of objects that map the attributes or field names of your Microsoft Teams content to Amazon Kendra index field names. For

Configuration	Description
• channelWiki • channelAttachment • meetingChat • meetingFile • meetingNote • calendarMeeting	more information, see [Mapping data source fields]().
additionalProperties	Additional configuration options for your content in your data source.
paymentModel	Specifies what type of payment model to use with your Microsoft Teams data source. Model A payment models are restricted to licensing and payment models that require security compliance. Model B payment models are suitable for licensing and payment models that do not require security compliance.
• inclusionTeamNameFilter • inclusionChannelNameFilter • inclusionFileNamePatterns • inclusionFileTypePatterns • inclusionUserEmailFilter • inclusionOneNoteSectionNamePatterns • inclusionOneNotePageNamePatterns	A list of regular expression patterns to *include* certain content in your Microsoft Teams data source. Content that matches the patterns are included in the index. Content that doesn't match the patterns are excluded from the index. If content matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the content isn't included in the index.

Configuration	Description
• exclusionTeamNameFilter • exclusionChannelNameFilter • exclusionFileNamePatterns • exclusionFileTypePatterns • exclusionUserEmailFilter • exclusionOneNoteSectionNamePatterns • exclusionOneNotePageNamePatterns	A list of regular expression patterns to *exclude* certain content in your Microsoft Teams data source. Content that matches the patterns are excluded from the index. Content that doesn't match the patterns are included in the index. If content matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the content isn't included in the index.
• isCrawlChatMessage • isCrawlChatAttachment • isCrawlChannelPost • isCrawlChannelAttachment • isCrawlChannelWiki • isCrawlCalendarMeeting • isCrawlMeetingChat • isCrawlMeetingFile • isCrawlMeetingNote	`true` to crawl these types of content in your Microsoft Teams data source.
startCalendarDateTime	You can configure a specific start date-time for your calendar content.
endCalendarDateTime	You can configure a specific end date-time for calendar content.
type	The type of data source. Specify `MSTEAMS` as your data source type.

Configuration	Description
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

Configuration	Description
secretArn	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Microsoft Teams. This includes your client ID and client secret that is generated when you create an OAuth application in the Azure portal.
version	The version of this template that is currently supported.

Microsoft Teams JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "tenantId": {
              "type": "string",
              "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]
{12}$",
              "minLength": 36,
              "maxLength": 36
            }
          },
          "required": [
            "tenantId"
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
```

```
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "chatMessage": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string",
                      "enum": [
                        "STRING",
                        "STRING_LIST",
                        "DATE"
                      ]
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
```

```
          ]
        },
        "chatAttachment": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string",
                      "enum": [
                        "STRING",
                        "DATE",
                        "LONG"
                      ]
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        },
        "channelPost": {
```

```
      "type": "object",
      "properties": {
        "fieldMappings": {
          "type": "array",
          "items": [
            {
              "type": "object",
              "properties": {
                "indexFieldName": {
                  "type": "string"
                },
                "indexFieldType": {
                  "type": "string",
                  "enum": [
                    "STRING",
                    "STRING_LIST",
                    "DATE"
                  ]
                },
                "dataSourceFieldName": {
                  "type": "string"
                },
                "dateFieldFormat": {
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                }
              },
              "required": [
                "indexFieldName",
                "indexFieldType",
                "dataSourceFieldName"
              ]
            }
          ]
        }
      },
      "required": [
        "fieldMappings"
      ]
    },
    "channelWiki": {
      "type": "object",
      "properties": {
        "fieldMappings": {
```

```
          "type": "array",
          "items": [
            {
              "type": "object",
              "properties": {
                "indexFieldName": {
                  "type": "string"
                },
                "indexFieldType": {
                  "type": "string",
                  "enum": [
                    "STRING",
                    "DATE",
                    "LONG"
                  ]
                },
                "dataSourceFieldName": {
                  "type": "string"
                },
                "dateFieldFormat": {
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                }
              },
              "required": [
                "indexFieldName",
                "indexFieldType",
                "dataSourceFieldName"
              ]
            }
          ]
        }
      },
      "required": [
        "fieldMappings"
      ]
    },
    "channelAttachment": {
      "type": "object",
      "properties": {
        "fieldMappings": {
          "type": "array",
          "items": [
            {
```

```
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
                    "enum": [
                      "STRING",
                      "DATE",
                      "LONG"
                    ]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  },
                  "dateFieldFormat": {
                    "type": "string",
                    "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                  }
                },
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        },
        "required": [
          "fieldMappings"
        ]
      },
      "meetingChat": {
        "type": "object",
        "properties": {
          "fieldMappings": {
            "type": "array",
            "items": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
```

```
                "type": "string"
              },
              "indexFieldType": {
                "type": "string",
                "enum": [
                  "STRING",
                  "STRING_LIST",
                  "DATE"
                ]
              },
              "dataSourceFieldName": {
                "type": "string"
              },
              "dateFieldFormat": {
                "type": "string",
                "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
              }
            },
            "required": [
              "indexFieldName",
              "indexFieldType",
              "dataSourceFieldName"
            ]
          }
        ]
      }
    },
    "required": [
      "fieldMappings"
    ]
  },
  "meetingFile": {
    "type": "object",
    "properties": {
      "fieldMappings": {
        "type": "array",
        "items": [
          {
            "type": "object",
            "properties": {
              "indexFieldName": {
                "type": "string"
              },
              "indexFieldType": {
```

```
                    "type": "string",
                    "enum": [
                      "STRING",
                      "DATE",
                      "LONG"
                    ]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  },
                  "dateFieldFormat": {
                    "type": "string",
                    "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                  }
                },
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        },
        "required": [
          "fieldMappings"
        ]
      },
      "meetingNote": {
        "type": "object",
        "properties": {
          "fieldMappings": {
            "type": "array",
            "items": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
                    "enum": [
                      "STRING",
```

```
                    "DATE"
                  ]
                },
                "dataSourceFieldName": {
                  "type": "string"
                },
                "dateFieldFormat": {
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                }
              },
              "required": [
                "indexFieldName",
                "indexFieldType",
                "dataSourceFieldName"
              ]
            }
          ]
        }
      },
      "required": [
        "fieldMappings"
      ]
    },
    "calendarMeeting": {
      "type": "object",
      "properties": {
        "fieldMappings": {
          "type": "array",
          "items": [
            {
              "type": "object",
              "properties": {
                "indexFieldName": {
                  "type": "string"
                },
                "indexFieldType": {
                  "type": "string",
                  "enum": [
                    "STRING",
                    "DATE"
                  ]
                },
                "dataSourceFieldName": {
```

```
                  "type": "string"
                },
                "dateFieldFormat": {
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                }
              },
              "required": [
                "indexFieldName",
                "indexFieldType",
                "dataSourceFieldName"
              ]
            }
          ]
        }
      },
      "required": [
        "fieldMappings"
      ]
    }
  }
},
 "additionalProperties": {
  "type": "object",
  "properties": {
    "paymentModel": {
      "type": "string",
      "enum": [
        "A",
        "B",
        "Evaluation Mode"
      ]
    },
    "inclusionTeamNameFilter": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "exclusionTeamNameFilter": {
      "type": "array",
      "items": {
        "type": "string"
      }
```

```
        },
        "inclusionChannelNameFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionChannelNameFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionFileNamePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionFileNamePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionFileTypePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionFileTypePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionUserEmailFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionOneNoteSectionNamePatterns": {
```

```
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "exclusionOneNoteSectionNamePatterns": {
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "inclusionOneNotePageNamePatterns": {
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "exclusionOneNotePageNamePatterns": {
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "isCrawlChatMessage": {
            "type": "boolean"
          },
          "isCrawlChatAttachment": {
            "type": "boolean"
          },
          "isCrawlChannelPost": {
            "type": "boolean"
          },
          "isCrawlChannelAttachment": {
            "type": "boolean"
          },
          "isCrawlChannelWiki": {
            "type": "boolean"
          },
          "isCrawlCalendarMeeting": {
            "type": "boolean"
          },
          "isCrawlMeetingChat": {
            "type": "boolean"
          },
```

```
        "isCrawlMeetingFile": {
          "type": "boolean"
        },
        "isCrawlMeetingNote": {
          "type": "boolean"
        },
        "startCalendarDateTime": {
          "anyOf": [
            {
              "type": "string",
              "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}Z$"
            },
            {
              "type": "string",
              "pattern": ""
            }
          ]
        },
        "endCalendarDateTime": {
          "anyOf": [
            {
              "type": "string",
              "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}Z$"
            },
            {
              "type": "string",
              "pattern": ""
            }
          ]
        }
      },
      "required": []
    },
    "type": {
      "type": "string",
      "pattern": "MSTEAMS"
    },
    "enableIdentityCrawler": {
      "type": "boolean"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
```

```
          "FULL_CRAWL",
          "CHANGE_LOG"
        ]
      },
      "secretArn": {
        "type": "string",
        "minLength": 20,
        "maxLength": 2048
      }
    },
    "version": {
      "type": "string",
      "anyOf": [
        {
          "pattern": "1.0.0"
        }
      ]
    },
    "required": [
      "connectionConfiguration",
      "repositoryConfigurations",
      "syncMode",
      "additionalProperties",
      "secretArn",
      "type"
    ]
}
```

Microsoft Yammer template schema

You include a JSON that contains the data source schema as part of TemplateConfiguration object. Specify the type of data source as `YAMMER`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the **Type** when you call CreateDataSource.

You can use the template provided in this developer guide.

The following table describes the parameters of the Microsoft Yammer JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the data source.

Configuration	Description
repositoryEndpointMetadata	The endpoint information for the data source. This data source does not specify an endpoint in `repositoryEndpointMetadata` . Rather, the connection information is included in an AWS Secrets Manager secret that you provide the `secretArn` .
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
• community • user • message • attachment	A list of objects that map attributes or field names of Microsoft Yammer content to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source
inclusionPatterns	A list of regular expression patterns to *include* certain files in your Microsoft Yammer data source. Files that match the patterns are included in the index. File that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedenc e and the file isn't included in the index.

Configuration	Description
exclusionPatterns	A list of regular expression patterns to *exclude* certain files in your Microsoft Yammer data source. Files that match the patterns are excluded from the index. Files that don't match the patterns are included in the index. If a file matches both an exclusion and inclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.
sinceDate	You can choose to configure a `sinceDate` parameter so that the Microsoft Yammer connector crawls content based on a specific `sinceDate` .
communityNameFilter	You can choose to index specific community content.
• isCrawlMessage • isCrawlAttachment • isCrawlPrivateMessage	`true` to crawl messages, message attachmen ts, and private messages.
type	Specify `YAMMER` as your data source type.
secretARN	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Microsoft Yammer. This includes your Microsoft Yammer user name and password, and client ID and client secret that is generated when you create an OAuth application in the Azure portal.

Configuration	Description
useChangeLog	`true` to use the Microsoft Yammer change log to determine which documents require updating in the index.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.

Microsoft Yammer JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
          }
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "community": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": {
                "anyOf": [
                  {
                    "type": "object",
                    "properties": {
                      "indexFieldName": {
                        "type": "string"
                      },
                      "indexFieldType": {
                        "type": "string",
                        "enum": [
                          "STRING",
                          "DATE"
                        ]
                      },
                      "dataSourceFieldName": {
                        "type": "string"
```

```
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          }
        },
        "required": [
          "fieldMappings"
        ]
      },
      "user": {
        "type": "object",
        "properties": {
          "fieldMappings": {
            "type": "array",
            "items": {
              "anyOf": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string",
                      "enum": [
                        "STRING",
                        "DATE"
                      ]
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
```

```
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                }
              },
              "required": [
                "indexFieldName",
                "indexFieldType",
                "dataSourceFieldName"
              ]
            }
          ]
        }
      }
    },
    "required": [
      "fieldMappings"
    ]
  },
  "message": {
    "type": "object",
    "properties": {
      "fieldMappings": {
        "type": "array",
        "items": {
          "anyOf": [
            {
              "type": "object",
              "properties": {
                "indexFieldName": {
                  "type": "string"
                },
                "indexFieldType": {
                  "type": "string",
                  "enum": [
                    "STRING",
                    "DATE"
                  ]
                },
                "dataSourceFieldName": {
                  "type": "string"
                },
                "dateFieldFormat": {
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
```

```
                  }
                },
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        }
      },
      "required": [
        "fieldMappings"
      ]
    },
    "attachment": {
      "type": "object",
      "properties": {
        "fieldMappings": {
          "type": "array",
          "items": {
            "anyOf": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
                    "enum": [
                      "STRING",
                      "DATE"
                    ]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  },
                  "dateFieldFormat": {
                    "type": "string",
                    "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                  }
                },
```

```
                    "required": [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
                    ]
                  }
                ]
              }
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      }
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "inclusionPatterns": {
          "type": "array"
        },
        "exclusionPatterns": {
          "type": "array"
        },
        "sinceDate": {
          "type": "string",
          "pattern": "^(19|2[0-9])[0-9]{2}-(0[1-9]|1[012])-(0[1-9]|[12][0-9]|
3[01])T(0[0-9]|1[0-9]|2[0-3]):([0-5][0-9]):([0-5][0-9])((\\+|-)(0[0-9]|1[0-9]|2[0-3]):
([0-5][0-9]))?$"
        },
        "communityNameFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "isCrawlMessage": {
          "type": "boolean"
        },
        "isCrawlAttachment": {
          "type": "boolean"
        },
        "isCrawlPrivateMessage": {
```

```
            "type": "boolean"
          }
        },
        "required": [
          "sinceDate"
        ]
      },
      "type": {
        "type": "string",
        "pattern": "YAMMER"
      },
      "secretArn": {
        "type": "string",
        "minLength": 20,
        "maxLength": 2048
      },
      "useChangeLog": {
        "type": "string",
        "enum": [
          "true",
          "false"
        ]
      },
      "syncMode": {
        "type": "string",
        "enum": [
          "FORCED_FULL_CRAWL",
          "FULL_CRAWL",
          "CHANGE_LOG"
        ]
      },
      "enableIdentityCrawler": {
        "type": "boolean"
      },
      "version": {
        "type": "string",
        "anyOf": [
          {
            "pattern": "1.0.0"
          }
        ]
      }
    },
    "required": [
```

```
    "connectionConfiguration",
    "repositoryConfigurations",
    "additionalProperties",
    "type",
    "secretArn",
    "syncMode"
  ]
}
```

MySQL template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as `JDBC`, the database type as `mysql`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See MySQL JSON schema.

The following table describes the parameters of the MySQL JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	Required configuration information for connecting your data source. • dbType—The type of Java database that you use, whether `mysql`, `db2`, `postgresql`, `oracle`, or `sqlserver`. • dbHost—The database host name. • dbPort—The database port. • dbInstance—The database instance.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.

Configuration	Description
	Specify the type of data source and the secret ARN.
document	A list of objects that map the attributes or field names of your database content to Amazon Kendra index field names. For more information, see [Mapping data source fields]().
additionalProperties	Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.
primaryKey	Provide the primary key for the database table. This identifies a table within your database.
titleColumn	Provide the name of the document title column within your database table.
bodyColumn	Provide the name of the document title column within your database table.
sqlQuery	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
timestampColumn	Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
timestampFormat	Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

Configuration	Description
timezone	Enter the name of the column which contains time zones for the content to be crawled.
changeDetectingColumns	Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns
allowedUsersColumns	Enter the name of the column which contains User IDs to be allowed access to content.
allowedGroupsColumn	Enter the name of the column which contains User IDs to be allowed access to content.
sourceURIColumn	Enter the name of the column which contains Source URLs to be indexed.
isSslEnabled	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
type	The type of data source. Specify `JDBC` as your data source type.

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

secretArn	The Amazon Resource Name (ARN) of a Secrets Manager secret that contains user name and password required to connect to your database. The secret must contain a JSON structure with the following keys:
	`{`
	`    "user name": "database user name",`
	`    "password": " password"`
	`}`

version	The version of the template that is currently supported.

MySQL JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "dbType": {
              "type": "string",
              "enum": [
                "mysql",
                "db2",
                "postgresql",
                "oracle",
                "sqlserver"
              ]
            },
            "dbHost": {
              "type": "string"
            },
            "dbPort": {
              "type": "string"
            },
            "dbInstance": {
              "type": "string"
            }
          },
          "required": [
            "dbType",
            "dbHost",
            "dbPort",
            "dbInstance"
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
```

```
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "document": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string"
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "primaryKey": {
          "type": "string"
```

```
        },
        "titleColumn": {
          "type": "string"
        },
        "bodyColumn": {
          "type": "string"
        },
        "sqlQuery": {
          "type": "string",
          "not": {
            "pattern": ";+"
          }
        },
        "timestampColumn": {
          "type": "string"
        },
        "timestampFormat": {
          "type": "string"
        },
        "timezone": {
          "type": "string"
        },
        "changeDetectingColumns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "allowedUsersColumn": {
          "type": "string"
        },
        "allowedGroupsColumn": {
          "type": "string"
        },
        "sourceURIColumn": {
          "type": "string"
        },
        "isSslEnabled": {
          "type": "boolean"
        }
      },
      "required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
    },
    "type" : {
```

```
        "type" : "string",
        "pattern": "JDBC"
      },
      "syncMode": {
        "type": "string",
        "enum": [
          "FORCED_FULL_CRAWL",
          "FULL_CRAWL",
          "CHANGE_LOG"
        ]
      },
      "secretArn": {
        "type": "string"
      }
    },
    "version": {
      "type": "string",
      "anyOf": [
        {
          "pattern": "1.0.0"
        }
      ]
    },
    "required": [
        "connectionConfiguration",
        "repositoryConfigurations",
        "syncMode",
        "additionalProperties",
        "secretArn",
        "type"
    ]
}
```

Oracle Database template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as `JDBC`, the database type as `oracle`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Oracle Database JSON schema.

The following table describes the parameters of the Oracle Database JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	Required configuration information for connecting your data source. • dbType—The type of Java database that you use, whether `mysql`, `db2`, `postgresql` , `oracle`, or `sqlserver` . • dbHost—The database host name. • dbPort—The database port. • dbInstance—The database instance.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings. Specify the type of data source and the secret ARN.
document	A list of objects that map the attributes or field names of your database content to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.
primaryKey	Provide the primary key for the database table. This identifies a table within your database.
titleColumn	Provide the name of the document title column within your database table.

Configuration	Description
bodyColumn	Provide the name of the document title column within your database table.
sqlQuery	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
timestampColumn	Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
timestampFormat	Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.
timezone	Enter the name of the column which contains time zones for the content to be crawled.
changeDetectingColumns	Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns
allowedUsersColumns	Enter the name of the column which contains User IDs to be allowed access to content.
allowedGroupsColumn	Enter the name of the column which contains User IDs to be allowed access to content.
sourceURIColumn	Enter the name of the column which contains Source URLs to be indexed.

Configuration	Description
isSslEnabled	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
type	The type of data source. Specify `JDBC` as your data source type.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

Configuration	Description
secretArn	The Amazon Resource Name (ARN) of a Secrets Manager secret that contains user name and password required to connect to your database. The secret must contain a JSON structure with the following keys: `{ "user name": "database user name", "password": " password" }`
version	The version of the template that is currently supported.

Oracle Database JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "dbType": {
              "type": "string",
              "enum": [
                "mysql",
                "db2",
                "postgresql",
                "oracle",
                "sqlserver"
              ]
            },
            "dbHost": {
              "type": "string"
```

```
          },
          "dbPort": {
            "type": "string"
          },
          "dbInstance": {
            "type": "string"
          }
        },
        "required": [
          "dbType",
          "dbHost",
          "dbPort",
          "dbInstance"
        ]
      }
    },
    "required": [
      "repositoryEndpointMetadata"
    ]
  },
  "repositoryConfigurations": {
    "type": "object",
    "properties": {
      "document": {
        "type": "object",
        "properties": {
          "fieldMappings": {
            "type": "array",
            "items": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string"
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  }
                },
                "required": [
                  "indexFieldName",
```

```
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "primaryKey": {
          "type": "string"
        },
        "titleColumn": {
          "type": "string"
        },
        "bodyColumn": {
          "type": "string"
        },
        "sqlQuery": {
          "type": "string",
          "not": {
            "pattern": ";+"
          }
        },
        "timestampColumn": {
          "type": "string"
        },
        "timestampFormat": {
          "type": "string"
        },
        "timezone": {
          "type": "string"
        },
        "changeDetectingColumns": {
          "type": "array",
```

```
          "items": {
            "type": "string"
          }
        },
        "allowedUsersColumn": {
          "type": "string"
        },
        "allowedGroupsColumn": {
          "type": "string"
        },
        "sourceURIColumn": {
          "type": "string"
        },
        "isSslEnabled": {
          "type": "boolean"
        }
      },
      "required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
    },
    "type" : {
      "type" : "string",
      "pattern": "JDBC"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL",
        "CHANGE_LOG"
      ]
    },
    "secretArn": {
      "type": "string"
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
```

```
        "connectionConfiguration",
        "repositoryConfigurations",
        "syncMode",
        "additionalProperties",
        "secretArn",
        "type"
    ]
}
```

PostgreSQL template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. Specify the type of data source as `JDBC`, the database type as `postgresql`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See PostgreSQL JSON schema.

The following table describes the parameters of the PostgreSQL JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	Required configuration information for connecting your data source. • dbType—The type of Java database that you use, whether `mysql`, `db2`, `postgresql`, `oracle`, or `sqlserver`. • dbHost—The database host name. • dbPort—The database port. • dbInstance—The database instance.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.

Configuration	Description
	Specify the type of data source and the secret ARN.
document	A list of objects that map the attributes or field names of your database content to Amazon Kendra index field names. For more information, see Mapping data source fields.
additionalProperties	Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.
primaryKey	Provide the primary key for the database table. This identifies a table within your database.
titleColumn	Provide the name of the document title column within your database table.
bodyColumn	Provide the name of the document title column within your database table.
sqlQuery	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
timestampColumn	Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
timestampFormat	Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

Configuration	Description
timezone	Enter the name of the column which contains time zones for the content to be crawled.
changeDetectingColumns	Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns
allowedUsersColumns	Enter the name of the column which contains User IDs to be allowed access to content.
allowedGroupsColumn	Enter the name of the column which contains User IDs to be allowed access to content.
sourceURIColumn	Enter the name of the column which contains Source URLs to be indexed.
isSslEnabled	Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
type	The type of data source. Specify `JDBC` as your data source type.

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

secretArn	The Amazon Resource Name (ARN) of a Secrets Manager secret that contains user name and password required to connect to your database. The secret must contain a JSON structure with the following keys:
	`{`
	`    "user name": "database user name",`
	`    "password": " password"`
	`}`

version	The version of the template that is currently supported.

PostgreSQL JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "dbType": {
              "type": "string",
              "enum": [
                "mysql",
                "db2",
                "postgresql",
                "oracle",
                "sqlserver"
              ]
            },
            "dbHost": {
              "type": "string"
            },
            "dbPort": {
              "type": "string"
            },
            "dbInstance": {
              "type": "string"
            }
          },
          "required": [
            "dbType",
            "dbHost",
            "dbPort",
            "dbInstance"
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
```

```
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "document": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string"
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      },
      "required": [
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "primaryKey": {
          "type": "string"
```

```
        },
        "titleColumn": {
          "type": "string"
        },
        "bodyColumn": {
          "type": "string"
        },
        "sqlQuery": {
          "type": "string",
          "not": {
            "pattern": ";+"
          }
        },
        "timestampColumn": {
          "type": "string"
        },
        "timestampFormat": {
          "type": "string"
        },
        "timezone": {
          "type": "string"
        },
        "changeDetectingColumns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "allowedUsersColumn": {
          "type": "string"
        },
        "allowedGroupsColumn": {
          "type": "string"
        },
        "sourceURIColumn": {
          "type": "string"
        },
        "isSslEnabled": {
          "type": "boolean"
        }
      },
      "required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
    },
    "type" : {
```

```
        "type" : "string",
        "pattern": "JDBC"
      },
      "syncMode": {
        "type": "string",
        "enum": [
          "FORCED_FULL_CRAWL",
          "FULL_CRAWL",
          "CHANGE_LOG"
        ]
      },
      "secretArn": {
        "type": "string"
      }
    },
    "version": {
      "type": "string",
      "anyOf": [
        {
          "pattern": "1.0.0"
        }
      ]
    },
    "required": [
        "connectionConfiguration",
        "repositoryConfigurations",
        "syncMode",
        "additionalProperties",
        "secretArn",
        "type"
    ]
}
```

Salesforce template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the Salesforce host URL as a part of the connection configuration or repository endpoint details. Also specify the type of data source as `SALESFORCEV2`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Salesforce JSON schema.

The following table describes the parameters of the Salesforce JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
hostUrl	The URL of the Salesforce instance to be indexed.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
• account • contact • campaign • case • product • lead • contract • partner • profile • idea • pricebook • task • solution • attachment • user • document • knowledgeArticles • group	A list of objects that map the attributes or field names of your Salesforce entities to Amazon Kendra index field names. For more information, see Mapping data source fields.

Configuration	Description
- opportunity - chatter - customEntity	
secretARN	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Salesforce. The secret must contain a JSON structure with the following keys: `{ "authenticationUrl": " OAUTH endpoint that Amazon Kendra connects to get an OAUTH token", "consumerKey": " Application public key generated when you created your Salesforce application ", "consumerSecret": " Application private key generated when you created your Salesforce application ", "password": " Password associated with the user logging in to the Salesforce instance ", "securityToken": " Token associated with the user account logging in to the Salesforce instance ", "username": " User name of the user logging in to the Salesforce instance" }`
additionalProperties	Additional configuration options for your content in your data source

Configuration	Description
- accountFilter - contactFilter - caseFilter - campaignFilter - contractFilter - groupFilter - leadFilter - productFilter - opportunityFilter - partnerFilter - pricebookFilter - ideaFilter - profileFilter - taskFilter - solutionFilter - userFilter - chatterFilter - documentFilter - knowledgeArticleFilter - customEntities	A collection of strings that specifies which entities to filter.

Configuration	Description
inclusionPatterns • inclusionDocumentFileTypePatterns • inclusionDocumentFileNamePatterns • inclusionAccountFileTypePatterns • inclusionCampaignFileTypePatterns • inclusionDocumentFileNamePatterns • inclusionCampaignFileNamePatterns • inclusionCaseFileTypePatterns • inclusionCaseFileNamePatterns • inclusionContactFileTypePatterns • inclusionContractFileNamePatterns • inclusionLeadFileTypePatterns • inclusionLeadFileNamePatterns • inclusionOpportunityFileTypePatterns • inclusionOpportunityFileNamePatterns • inclusionSolutionFileTypePatterns • inclusionSolutionFileNamePatterns • inclusionTaskFileTypePatterns • inclusionTaskFileNamePatterns • inclusionGroupFileTypePatterns • inclusionGroupFileNamePatterns • inclusionChatterFileTypePatterns • inclusionChatterFileNamePatterns • inclusionCustomEntityFileTypePatterns • inclusionCustomEntityFileNamePatterns	A list of regular expression patterns to *include* certain files in your Salesforce data source. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.

Configuration	Description
exclusionPatterns • exclusionDocumentFileTypePatterns • exclusionDocumentFileNamePatterns • exclusionAccountFileTypePatterns • exclusionCampaignFileTypePatterns • exclusionCampaignFileNamePatterns • exclusionCaseFileTypePatterns • exclusionCaseFileNamePatterns • exclusionContactFileTypePatterns • exclusionContractFileNamePatterns • exclusionLeadFileTypePatterns • exclusionLeadFileNamePatterns • exclusionOpportunityFileTypePatterns • exclusionOpportunityFileNamePatterns • exclusionSolutionFileTypePatterns • exclusionSolutionFileNamePatterns • exclusionTaskFileTypePatterns • exclusionTaskFileNamePatterns • exclusionGroupFileTypePatterns • exclusionGroupFileNamePatterns • exclusionChatterFileTypePatterns • exclusionChatterFileNamePatterns • exclusionCustomEntityFileTypePatterns • exclusionCustomEntityFileNamePatterns	A list of regular expression patterns to *exclude* certain files in your Salesforce data source. Files that match the patterns are excluded from the index. Files that don't match the patterns are included in the index. If a file matches both an exclusion and inclusion pattern, the exclusion pattern takes precedenc e and the file isn't included in the index.

Configuration	Description
- isCrawlAccount - isCrawlContact - isCrawlCase - isCrawlCampaign - isCrawlProduct - isCrawlLead - isCrawlContract - isCrawlPartner - isCrawlProfile - isCrawlIdea - isCrawlPricebook - isCrawlDocument - crawlSharedDocument - isCrawlGroup - isCrawlOpportunity - isCrawlChatter - isCrawlUser - isCrawlSolution - isCrawlTask - isCrawlAccountAttachments - isCrawlContactAttachments - isCrawlCaseAttachments - isCrawlCampaignAttachments - isCrawlLeadAttachments - isCrawlContractAttachments - isCrawlGroupAttachments - isCrawlOpportunityAttachments - isCrawlChatterAttachments - isCrawlSolutionAttachments	`true` to crawl these types of files in your Salesforce account.

Configuration	Description

• isCrawlTaskAttachments
• isCrawlCustomEntityAttachments
• isCrawlKnowledgeArticles
  • isCrawlDraft
  • isCrawlPublish
  • isCrawlArchived	

type	The type of data source. Specify `SALESFORCEV2` as your data source type.
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.

Configuration	Description
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
version	The version of this template that is currently supported.

Salesforce JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties":
  {
    "connectionConfiguration": {
      "type": "object",
      "properties":
      {
```

```
        "repositoryEndpointMetadata":
        {
          "type": "object",
          "properties":
          {
            "hostUrl":
            {
              "type": "string",
              "pattern": "https:.*"
            }
          },
          "required":
          [
            "hostUrl"
          ]
        }
      },
      "required":
      [
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties":
      {
        "account":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
                {
                  "type": "object",
                  "properties":
                  {
                    "indexFieldName":
                    {
                      "type": "string"
                    },
```

```
                    "indexFieldType":
                    {
                      "type": "string",
                      "enum":
                      [
                        "STRING",
                        "STRING_LIST",
                        "DATE",
                        "LONG"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
                    {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required":
                  [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required":
          [
            "fieldMappings"
          ]
        },
        "contact":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
```

```
                "items":
                [
                  {
                    "type": "object",
                    "properties":
                    {
                      "indexFieldName":
                      {
                        "type": "string"
                      },
                      "indexFieldType":
                      {
                        "type": "string",
                        "enum":
                        [
                          "STRING",
                          "STRING_LIST",
                          "DATE"
                        ]
                      },
                      "dataSourceFieldName":
                      {
                        "type": "string"
                      },
                      "dateFieldFormat":
                      {
                        "type": "string",
                        "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                      }
                    },
                    "required":
                    [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
                    ]
                  }
                ]
              }
            },
            "required":
            [
              "fieldMappings"
            ]
```

```
        },
        "campaign":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
                {
                  "type": "object",
                  "properties":
                  {
                    "indexFieldName":
                    {
                      "type": "string"
                    },
                    "indexFieldType":
                    {
                      "type": "string",
                      "enum":
                      [
                        "STRING",
                        "STRING_LIST",
                        "DATE",
                        "LONG"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
                    {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required":
                  [
                    "indexFieldName",
                    "indexFieldType",
```

```
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required":
          [
            "fieldMappings"
          ]
        },
        "case":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
                {
                  "type": "object",
                  "properties":
                  {
                    "indexFieldName":
                    {
                      "type": "string"
                    },
                    "indexFieldType":
                    {
                      "type": "string",
                      "enum":
                      [
                        "STRING",
                        "STRING_LIST",
                        "DATE"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
```

```
                {
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                }
              },
              "required":
              [
                "indexFieldName",
                "indexFieldType",
                "dataSourceFieldName"
              ]
            }
          ]
        }
      },
      "required":
      [
        "fieldMappings"
      ]
    },
    "product":
    {
      "type": "object",
      "properties":
      {
        "fieldMappings":
        {
          "type": "array",
          "items":
          [
            {
              "type": "object",
              "properties":
              {
                "indexFieldName":
                {
                  "type": "string"
                },
                "indexFieldType":
                {
                  "type": "string",
                  "enum":
                  [
                    "STRING",
```

```
                        "STRING_LIST",
                        "DATE"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
                    {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required":
                  [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required":
          [
            "fieldMappings"
          ]
        },
        "lead":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
                {
                  "type": "object",
                  "properties":
                  {
                    "indexFieldName":
```

```
                    {
                      "type": "string"
                    },
                    "indexFieldType":
                    {
                      "type": "string",
                      "enum":
                      [
                        "STRING",
                        "STRING_LIST",
                        "DATE",
                        "LONG"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
                    {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required":
                  [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required":
          [
            "fieldMappings"
          ]
        },
        "contract":
        {
          "type": "object",
          "properties":
          {
```

```
"fieldMappings":
{
  "type": "array",
  "items":
  [
    {
      "type": "object",
      "properties":
      {
        "indexFieldName":
        {
          "type": "string"
        },
        "indexFieldType":
        {
          "type": "string",
          "enum":
          [
            "STRING",
            "STRING_LIST",
            "DATE"
          ]
        },
        "dataSourceFieldName":
        {
          "type": "string"
        },
        "dateFieldFormat":
        {
          "type": "string",
          "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
        }
      },
      "required":
      [
        "indexFieldName",
        "indexFieldType",
        "dataSourceFieldName"
      ]
    }
  ]
}
},
"required":
```

```
            [
              "fieldMappings"
            ]
          },
          "partner":
          {
            "type": "object",
            "properties":
            {
              "fieldMappings":
              {
                "type": "array",
                "items":
                [
                  {
                    "type": "object",
                    "properties":
                    {
                      "indexFieldName":
                      {
                        "type": "string"
                      },
                      "indexFieldType":
                      {
                        "type": "string",
                        "enum":
                        [
                          "STRING",
                          "STRING_LIST",
                          "DATE"
                        ]
                      },
                      "dataSourceFieldName":
                      {
                        "type": "string"
                      },
                      "dateFieldFormat":
                      {
                        "type": "string",
                        "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                      }
                    },
                    "required":
                    [
```

```
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required":
          [
            "fieldMappings"
          ]
        },
        "profile":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
                {
                  "type": "object",
                  "properties":
                  {
                    "indexFieldName":
                    {
                      "type": "string"
                    },
                    "indexFieldType":
                    {
                      "type": "string",
                      "enum":
                      [
                        "STRING",
                        "STRING_LIST",
                        "DATE"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
```

```
                    },
                    "dateFieldFormat":
                    {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required":
                  [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required":
          [
            "fieldMappings"
          ]
        },
        "idea":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
                {
                  "type": "object",
                  "properties":
                  {
                    "indexFieldName":
                    {
                      "type": "string"
                    },
                    "indexFieldType":
                    {
                      "type": "string",
                      "enum":
```

```
                      [
                        "STRING",
                        "STRING_LIST",
                        "DATE",
                        "LONG"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
                    {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required":
                  [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required":
          [
            "fieldMappings"
          ]
        },
        "pricebook":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
                {
                  "type": "object",
```

```
                   "properties":
                   {
                     "indexFieldName":
                     {
                       "type": "string"
                     },
                     "indexFieldType":
                     {
                       "type": "string",
                       "enum":
                       [
                         "STRING",
                         "STRING_LIST",
                         "DATE"
                       ]
                     },
                     "dataSourceFieldName":
                     {
                       "type": "string"
                     },
                     "dateFieldFormat":
                     {
                       "type": "string",
                       "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                     }
                   },
                   "required":
                   [
                     "indexFieldName",
                     "indexFieldType",
                     "dataSourceFieldName"
                   ]
                 }
               ]
             }
           },
           "required":
           [
             "fieldMappings"
           ]
         },
         "task":
         {
           "type": "object",
```

```
            "properties":
            {
              "fieldMappings":
              {
                "type": "array",
                "items":
                [
                  {
                    "type": "object",
                    "properties":
                    {
                      "indexFieldName":
                      {
                        "type": "string"
                      },
                      "indexFieldType":
                      {
                        "type": "string",
                        "enum":
                        [
                          "STRING",
                          "STRING_LIST",
                          "DATE"
                        ]
                      },
                      "dataSourceFieldName":
                      {
                        "type": "string"
                      },
                      "dateFieldFormat":
                      {
                        "type": "string",
                        "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                      }
                    },
                    "required":
                    [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
                    ]
                  }
                ]
              }
```

```
      },
      "required":
      [
        "fieldMappings"
      ]
    },
    "solution":
    {
      "type": "object",
      "properties":
      {
        "fieldMappings":
        {
          "type": "array",
          "items":
          [
            {
              "type": "object",
              "properties":
              {
                "indexFieldName":
                {
                  "type": "string"
                },
                "indexFieldType":
                {
                  "type": "string",
                  "enum":
                  [
                    "STRING",
                    "STRING_LIST",
                    "DATE"
                  ]
                },
                "dataSourceFieldName":
                {
                  "type": "string"
                },
                "dateFieldFormat":
                {
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                }
              },
```

```
                    "required":
                    [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
                    ]
                  }
                ]
              }
            },
            "required":
            [
              "fieldMappings"
            ]
          },
          "attachment":
          {
            "type": "object",
            "properties":
            {
              "fieldMappings":
              {
                "type": "array",
                "items":
                [
                  {
                    "type": "object",
                    "properties":
                    {
                      "indexFieldName":
                      {
                        "type": "string"
                      },
                      "indexFieldType":
                      {
                        "type": "string",
                        "enum":
                        [
                          "STRING",
                          "STRING_LIST",
                          "DATE",
                          "LONG"
                        ]
                      },
```

```
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
                    {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required":
                  [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required":
          [
            "fieldMappings"
          ]
        },
        "user":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
                {
                  "type": "object",
                  "properties":
                  {
                    "indexFieldName":
                    {
                      "type": "string"
                    },
                    "indexFieldType":
```

```
                    {
                      "type": "string",
                      "enum":
                      [
                        "STRING",
                        "STRING_LIST",
                        "DATE"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
                    {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required":
                  [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required":
          [
            "fieldMappings"
          ]
        },
        "document":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
```

```
                {
                  "type": "object",
                  "properties":
                  {
                    "indexFieldName":
                    {
                      "type": "string"
                    },
                    "indexFieldType":
                    {
                      "type": "string",
                      "enum":
                      [
                        "STRING",
                        "STRING_LIST",
                        "DATE",
                        "LONG"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
                    {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required":
                  [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required":
          [
            "fieldMappings"
          ]
        },
```

```
        "knowledgeArticles":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
                {
                  "type": "object",
                  "properties":
                  {
                    "indexFieldName":
                    {
                      "type": "string"
                    },
                    "indexFieldType":
                    {
                      "type": "string",
                      "enum":
                      [
                        "STRING",
                        "STRING_LIST",
                        "DATE"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
                    {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required":
                  [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
```

```
                }
              ]
            }
          },
          "required":
          [
            "fieldMappings"
          ]
        },
        "group":
        {
          "type": "object",
          "properties":
          {
            "fieldMappings":
            {
              "type": "array",
              "items":
              [
                {
                  "type": "object",
                  "properties":
                  {
                    "indexFieldName":
                    {
                      "type": "string"
                    },
                    "indexFieldType":
                    {
                      "type": "string",
                      "enum":
                      [
                        "STRING",
                        "STRING_LIST",
                        "DATE"
                      ]
                    },
                    "dataSourceFieldName":
                    {
                      "type": "string"
                    },
                    "dateFieldFormat":
                    {
                      "type": "string",
```

```
                    "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                  }
                },
                "required":
                [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        },
        "required":
        [
          "fieldMappings"
        ]
      },
      "opportunity":
      {
        "type": "object",
        "properties":
        {
          "fieldMappings":
          {
            "type": "array",
            "items":
            [
              {
                "type": "object",
                "properties":
                {
                  "indexFieldName":
                  {
                    "type": "string"
                  },
                  "indexFieldType":
                  {
                    "type": "string",
                    "enum":
                    [
                      "STRING",
                      "STRING_LIST",
                      "DATE",
```

```
                    "LONG"
                  ]
                },
                "dataSourceFieldName":
                {
                  "type": "string"
                },
                "dateFieldFormat":
                {
                  "type": "string",
                  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                }
              },
              "required":
              [
                "indexFieldName",
                "indexFieldType",
                "dataSourceFieldName"
              ]
            }
          ]
        }
      },
      "required":
      [
        "fieldMappings"
      ]
    },
    "chatter":
    {
      "type": "object",
      "properties":
      {
        "fieldMappings":
        {
          "type": "array",
          "items":
          [
            {
              "type": "object",
              "properties":
              {
                "indexFieldName":
                {
```

```
                    "type": "string"
                  },
                  "indexFieldType":
                  {
                    "type": "string",
                    "enum":
                    [
                      "STRING",
                      "STRING_LIST",
                      "DATE"
                    ]
                  },
                  "dataSourceFieldName":
                  {
                    "type": "string"
                  },
                  "dateFieldFormat":
                  {
                    "type": "string",
                    "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                  }
                },
                "required":
                [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        },
        "required":
        [
          "fieldMappings"
        ]
      },
      "customEntity":
      {
        "type": "object",
        "properties":
        {
          "fieldMappings":
          {
```

```
                "type": "array",
                "items":
                [
                  {
                    "type": "object",
                    "properties":
                    {
                      "indexFieldName":
                      {
                        "type": "string"
                      },
                      "indexFieldType":
                      {
                        "type": "string",
                        "enum":
                        [
                          "STRING",
                          "STRING_LIST",
                          "DATE"
                        ]
                      },
                      "dataSourceFieldName":
                      {
                        "type": "string"
                      },
                      "dateFieldFormat":
                      {
                        "type": "string",
                        "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                      }
                    },
                    "required":
                    [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
                    ]
                  }
                ]
              }
            },
            "required":
            [
              "fieldMappings"
```

```
          ]
        }
      }
    },
    "additionalProperties": {
      "type": "object",
      "properties":
      {
        "accountFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "contactFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "caseFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "campaignFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "contractFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "groupFilter":{
```

```
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "leadFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "productFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "opportunityFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "partnerFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "pricebookFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "ideaFilter":{
          "type": "array",
          "items":
```

```
          {
            "type": "string"
          }
        },
        "profileFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "taskFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "solutionFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "userFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "chatterFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "documentFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
```

```
          }
        },
        "knowledgeArticleFilter":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "customEntities":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "isCrawlAccount": {
          "type": "boolean"
        },
        "isCrawlContact": {
          "type": "boolean"
        },
        "isCrawlCase": {
          "type": "boolean"
        },
        "isCrawlCampaign": {
          "type": "boolean"
        },
        "isCrawlProduct": {
          "type": "boolean"
        },
        "isCrawlLead": {
          "type": "boolean"
        },
        "isCrawlContract": {
          "type": "boolean"
        },
        "isCrawlPartner": {
          "type": "boolean"
        },
        "isCrawlProfile": {
          "type": "boolean"
        },
        "isCrawlIdea": {
```

```
            "type": "boolean"
          },
          "isCrawlPricebook": {
            "type": "boolean"
          },
          "isCrawlDocument": {
            "type": "boolean"
          },
          "crawlSharedDocument": {
            "type": "boolean"
          },
          "isCrawlGroup": {
            "type": "boolean"
          },
          "isCrawlOpportunity": {
            "type": "boolean"
          },
          "isCrawlChatter": {
            "type": "boolean"
          },
          "isCrawlUser": {
            "type": "boolean"
          },
          "isCrawlSolution":{
            "type": "boolean"
          },
          "isCrawlTask":{
            "type": "boolean"
          },

          "isCrawlAccountAttachments": {
            "type": "boolean"
          },
          "isCrawlContactAttachments": {
            "type": "boolean"
          },
          "isCrawlCaseAttachments": {
            "type": "boolean"
          },
          "isCrawlCampaignAttachments": {
            "type": "boolean"
          },
          "isCrawlLeadAttachments": {
            "type": "boolean"
```

```
        },
        "isCrawlContractAttachments": {
          "type": "boolean"
        },
        "isCrawlGroupAttachments": {
          "type": "boolean"
        },
        "isCrawlOpportunityAttachments": {
          "type": "boolean"
        },
        "isCrawlChatterAttachments": {
          "type": "boolean"
        },
        "isCrawlSolutionAttachments":{
          "type": "boolean"
        },
        "isCrawlTaskAttachments":{
          "type": "boolean"
        },
        "isCrawlCustomEntityAttachments":{
          "type": "boolean"
        },
        "isCrawlKnowledgeArticles": {
          "type": "object",
          "properties":
          {
            "isCrawlDraft": {
              "type": "boolean"
            },
            "isCrawlPublish": {
              "type": "boolean"
            },
            "isCrawlArchived": {
              "type": "boolean"
            }
          }
        },
        "inclusionDocumentFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
```

```
          "exclusionDocumentFileTypePatterns": {
            "type": "array",
            "items":
            {
              "type": "string"
            }
          },
          "inclusionDocumentFileNamePatterns":{
            "type": "array",
            "items":
            {
              "type": "string"
            }
          },
          "exclusionDocumentFileNamePatterns": {
            "type": "array",
            "items":
            {
              "type": "string"
            }
          },
          "inclusionAccountFileTypePatterns": {
            "type": "array",
            "items":
            {
              "type": "string"
            }
          },
          "exclusionAccountFileTypePatterns": {
            "type": "array",
            "items":
            {
              "type": "string"
            }
          },
          "inclusionAccountFileNamePatterns":{
            "type": "array",
            "items":
            {
              "type": "string"
            }
          },
          "exclusionAccountFileNamePatterns":{
            "type": "array",
```

```
          "items":
          {
            "type": "string"
          }
        },
        "inclusionCampaignFileTypePatterns": {
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionCampaignFileTypePatterns": {
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionCampaignFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionCampaignFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionCaseFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionCaseFileTypePatterns":{
          "type": "array",
          "items":
          {
```

```
            "type": "string"
          }
        },
        "inclusionCaseFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionCaseFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionContactFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionContactFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionContactFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionContactFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
```

```
        },
        "inclusionContractFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionContractFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionContractFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionContractFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionLeadFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionLeadFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionLeadFileNamePatterns":{
```

```
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionLeadFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionOpportunityFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionOpportunityFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionOpportunityFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionOpportunityFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionSolutionFileTypePatterns":{
          "type": "array",
          "items":
```

```
            {
              "type": "string"
            }
          },
          "exclusionSolutionFileTypePatterns":{
            "type": "array",
            "items":
            {
              "type": "string"
            }
          },
          "inclusionSolutionFileNamePatterns":{
            "type": "array",
            "items":
            {
              "type": "string"
            }
          },
          "exclusionSolutionFileNamePatterns":{
            "type": "array",
            "items":
            {
              "type": "string"
            }
          },
          "inclusionTaskFileTypePatterns":{
            "type": "array",
            "items":
            {
              "type": "string"
            }
          },
          "exclusionTaskFileTypePatterns":{
            "type": "array",
            "items":
            {
              "type": "string"
            }
          },
          "inclusionTaskFileNamePatterns":{
            "type": "array",
            "items":
            {
              "type": "string"
```

```
        }
      },
      "exclusionTaskFileNamePatterns":{
        "type": "array",
        "items":
        {
          "type": "string"
        }
      },
      "inclusionGroupFileTypePatterns":{
        "type": "array",
        "items":
        {
          "type": "string"
        }
      },
      "exclusionGroupFileTypePatterns":{
        "type": "array",
        "items":
        {
          "type": "string"
        }
      },
      "inclusionGroupFileNamePatterns":{
        "type": "array",
        "items":
        {
          "type": "string"
        }
      },
      "exclusionGroupFileNamePatterns":{
        "type": "array",
        "items":
        {
          "type": "string"
        }
      },
      "inclusionChatterFileTypePatterns":{
        "type": "array",
        "items":
        {
          "type": "string"
        }
      },
```

```
        "exclusionChatterFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionChatterFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionChatterFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionCustomEntityFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionCustomEntityFileTypePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "inclusionCustomEntityFileNamePatterns":{
          "type": "array",
          "items":
          {
            "type": "string"
          }
        },
        "exclusionCustomEntityFileNamePatterns":{
          "type": "array",
```

```
          "items":
          {
            "type": "string"
          }
        }
      },
      "required":
      []
    },
    "enableIdentityCrawler": {
      "type": "boolean"
    },
    "type": {
      "type": "string",
      "pattern": "SALESFORCEV2"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FULL_CRAWL",
        "FORCED_FULL_CRAWL",
        "CHANGE_LOG"
      ]
    },
    "secretArn": {
      "type": "string",
      "minLength": 20,
      "maxLength": 2048
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
    "connectionConfiguration",
    "repositoryConfigurations",
    "syncMode",
    "additionalProperties",
    "secretArn",
```

```
    "type"
  ]
}
```

ServiceNow template schema

You include a JSON that contains the data source schema as part of the TemplateConfiguration object. You provide the ServiceNow host URL, authentication type, and instance version as a part of the connection configuration or repository endpoint details. Also specify the type of data source as `SERVICENOWV2`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See ServiceNow JSON schema.

The following table describes the parameters of the ServiceNow JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
hostUrl	The ServiceNow host URL. For example, *`your-domain.service-now.com`*.
authType	The type of authentication that you use, whether `basicAuth` or `OAuth2`.
servicenowInstanceVersion	The ServiceNow version that you use. You can choose between `Tokyo`, `Sandiego`, `Rome`, and `Others`.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
• knowledgeArticle • attachment • serviceCatalog	A list of objects that map the attributes or field names of your ServiceNow knowledge articles, attachments, service catalog, and incidents to Amazon Kendra index field names. For more information, see Mapping data

Configuration	Description
• incident	source fields. The ServiceNow data source field names must exist in your ServiceNow custom metadata.
additional properties	Additional configuration options for your content in your data source.
maxFileSizeInMegaBytes	Specify the file size limit in MBs that Amazon Kendra will crawl. Amazon Kendra will crawl only the files within the size limit you define. The default file size is 50MB. The maximum file size should be greater than 0MB and less than or equal to 50MB.
• knowledgeArticleFilter • incidentQueryFilter • serviceCatalogQueryFilter • knowledgeArticleTitleRegExp • serviceCatalogTitleRegExp • incidentTitleRegExp • inclusionFileTypePatterns • exclusionFileTypePatterns • inclusionFileNamePatterns • exclusionFileNamePatterns • incidentStateType	A list of regular expression patterns to include and/or exclude certain files in your ServiceNow data source. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.

Configuration	Description
- isCrawlKnowledgeArticle - isCrawlKnowledgeArticleAttachment - includePublicArticlesOnly - isCrawlServiceCatalog - isCrawlServiceCatalogAttachment - isCrawlActiveServiceCatalog - isCrawlInactiveServiceCatalog - isCrawlIncident - isCrawlIncidentAttachment - isCrawlActiveIncident - isCrawlInactiveIncident - applyACLForKnowledgeArticle - applyACLForServiceCatalog - applyACLForIncident	`true` to crawl ServiceNow knowledge articles, service catalogs, incidents, and attachments.
type	The type of data source. Specify `SERVICENOWV2` as your data source type.
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.

<table>
<tr><th>Configuration</th><th>Description</th></tr>
<tr><td>syncMode</td><td>Specify how Amazon Kendra should update your index when your data source content changes. You can choose between:

<code>FORCED_FULL_CRAWL</code> to freshly index all content, replacing existing content each time your data source syncs with your index.
<code>FULL_CRAWL</code> to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
</td></tr>
<tr><td>secretARN</td><td>The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your ServiceNow. The secret must contain a JSON structure with the following keys:
<pre>{
    "username": " user name",
    "password": " password"
}</pre>
If you use OAuth2 authentication, your secret must contain a JSON structure with the following keys:
<pre>{
    "username": " user name",
    "password": " password",
    "clientId": " client id",
    "clientSecret": " client secret"
}</pre></td></tr>
<tr><td>version</td><td>The version of the template that is currently supported.</td></tr>
</table>

ServiceNow JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "hostUrl": {
              "type": "string",
              "pattern": "^(?!(^(https?|ftp|file):\/\/))[a-z0-9-]+(.service-
now.com|.servicenowservices.com)$",
              "minLength": 1,
              "maxLength": 2048
            },
            "authType": {
              "type": "string",
              "enum": [
                "basicAuth",
                "OAuth2"
              ]
            },
            "servicenowInstanceVersion": {
              "type": "string",
              "enum": [
                "Tokyo",
                "Sandiego",
                "Rome",
                "Others"
                ]
            }
          },
          "required": [
            "hostUrl",
            "authType",
            "servicenowInstanceVersion"
          ]
        }
      },
      "required": [
```

```
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "knowledgeArticle": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string",
                      "enum": [
                        "STRING",
                        "DATE",
                        "STRING_LIST"
                      ]
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
```

```
              "fieldMappings"
            ]
          },
          "attachment": {
            "type": "object",
            "properties": {
              "fieldMappings": {
                "type": "array",
                "items": [
                  {
                    "type": "object",
                    "properties": {
                      "indexFieldName": {
                        "type": "string"
                      },
                      "indexFieldType": {
                        "type": "string",
                        "enum": [
                          "STRING",
                          "LONG",
                          "DATE",
                          "STRING_LIST"
                        ]
                      },
                      "dataSourceFieldName": {
                        "type": "string"
                      },
                      "dateFieldFormat": {
                        "type": "string",
                        "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                      }
                    },
                    "required": [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
                    ]
                  }
                ]
              }
            },
            "required": [
              "fieldMappings"
            ]
```

```
        },
        "serviceCatalog": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string",
                      "enum": [
                        "STRING",
                        "DATE",
                        "STRING_LIST"
                      ]
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        },
        "incident": {
          "type": "object",
```

```
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "indexFieldName": {
                      "type": "string"
                    },
                    "indexFieldType": {
                      "type": "string",
                      "enum": [
                        "STRING",
                        "DATE",
                        "STRING_LIST"
                      ]
                    },
                    "dataSourceFieldName": {
                      "type": "string"
                    },
                    "dateFieldFormat": {
                      "type": "string",
                      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                    }
                  },
                  "required": [
                    "indexFieldName",
                    "indexFieldType",
                    "dataSourceFieldName"
                  ]
                }
              ]
            }
          },
          "required": [
            "fieldMappings"
          ]
        }
      }
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
```

```
        "maxFileSizeInMegaBytes": {
          "type": "string"
        },
        "isCrawlKnowledgeArticle": {
          "type": "boolean"
        },
        "isCrawlKnowledgeArticleAttachment": {
          "type": "boolean"
        },
        "includePublicArticlesOnly": {
          "type": "boolean"
        },
        "knowledgeArticleFilter": {
          "type": "string"
        },
        "incidentQueryFilter": {
          "type": "string"
        },
        "serviceCatalogQueryFilter": {
          "type": "string"
        },
        "isCrawlServiceCatalog": {
          "type": "boolean"
        },
        "isCrawlServiceCatalogAttachment": {
          "type": "boolean"
        },
        "isCrawlActiveServiceCatalog": {
          "type": "boolean"
        },
        "isCrawlInactiveServiceCatalog": {
          "type": "boolean"
        },
        "isCrawlIncident": {
          "type": "boolean"
        },
        "isCrawlIncidentAttachment": {
          "type": "boolean"
        },
        "isCrawlActiveIncident": {
          "type": "boolean"
        },
        "isCrawlInactiveIncident": {
          "type": "boolean"
```

```
        },
        "applyACLForKnowledgeArticle": {
          "type": "boolean"
        },
        "applyACLForServiceCatalog": {
          "type": "boolean"
        },
        "applyACLForIncident": {
          "type": "boolean"
        },
        "incidentStateType": {
          "type": "array",
          "items": {
            "type": "string",
            "enum": [
              "Open",
              "Open - Unassigned",
              "Resolved",
              "All"
            ]
          }
        },
        "knowledgeArticleTitleRegExp": {
          "type": "string"
        },
        "serviceCatalogTitleRegExp": {
          "type": "string"
        },
        "incidentTitleRegExp": {
          "type": "string"
        },
        "inclusionFileTypePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionFileTypePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionFileNamePatterns": {
```

```
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "exclusionFileNamePatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      },
      "required": []
    },
    "type": {
      "type": "string",
      "pattern": "SERVICENOWV2"
    },
    "enableIdentityCrawler": {
      "type": "boolean"
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL"
      ]
    },
    "secretArn": {
      "type": "string",
      "minLength": 20,
      "maxLength": 2048
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
  },
  "required": [
    "connectionConfiguration",
```

```
    "repositoryConfigurations",
    "syncMode",
    "additionalProperties",
    "secretArn",
    "type"
  ]
}
```

Slack template schema

You include a JSON that contains the data source schema as part of TemplateConfiguration object. You provide the host URL as a part of the connection configuration or repository endpoint details. Also specify the type of data source as `SLACK`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Slack JSON schema.

The following table describes the parameters of the Slack JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
teamId	The Slack team ID you copied from your Slack main page URL.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
All	A list of objects that map the attributes or field names of your Slack content to Amazon Kendra index field names.
additionalProperties	Additional configuration options for your content in your data source.

Configuration	Description
inclusionPatterns	A list of regular expression patterns to include specific content in your Slack data source. Content that matches the patterns are included in the index. Content that doesn't match the patterns are excluded from the index. If any content matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the content isn't included in the index.
exclusionPatterns	A list of regular expression patterns to exclude specific content in your Slack data source. Content that matches the patterns are excluded from the index. Content that doen't match the patterns are included in the index. If any content matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the content isn't included in the index.
crawlBotMessages	`true` to crawl bot messages.
excludeArchived	`true` to exclude crawling of archived messages.
conversationType	The type of conversation that you want to index whether `PUBLIC_CHANNEL` , `PRIVATE_CHANNEL` , `GROUP_MESSAGE` and `DIRECT_MESSAGE` .
channelFilter	The type of channel that you want to index whether `private_channel` or `public_channel` .

Configuration	Description
sinceDate	You can choose to configure a `sinceDate` parameter so that the Slack connector crawls content based on a specific `sinceDate` .
lookBack	You can choose to configure a `lookBack` parameter so that the Slack connector crawls updated or deleted content upto a specified number of hours before your last connector sync.
syncMode	Specify how Amazon Kendra should update your index when your data source content changes. You can choose between: • `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index. • `FULL_CRAWL` to index only new, modified and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync. • `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
type	The type of data source. Specify `SLACK` as your data source type.

Configuration	Description
enableIdentityCrawler	`true` to use Amazon Kendra's identity crawler to sync identity/principal information on users and groups with access to certain documents. If identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information.
secretArn	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Slack. The secret must contain a JSON structure with the following keys: `{ "slackToken": " token" }`
version	The version of this template that's currently supported.

Slack JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "teamId": {
```

```
                "type": "string"
              }
            },
            "required": ["teamId"]
          }
        }
      },
      "repositoryConfigurations": {
        "type": "object",
        "properties": {
          "All": {
            "type": "object",
            "properties": {
              "fieldMappings": {
                "type": "array",
                "items": [
                  {
                    "type": "object",
                    "properties": {
                      "indexFieldName": {
                        "type": "string"
                      },
                      "indexFieldType": {
                        "type": "string",
                        "enum": ["STRING", "STRING_LIST", "DATE","LONG"]
                      },
                      "dataSourceFieldName": {
                        "type": "string"
                      },
                      "dateFieldFormat": {
                        "type": "string",
                        "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
                      }
                    },
                    "required": [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
                    ]
                  }
                ]
              }
            },
            "required": [
```

```
            "fieldMappings"
          ]
        }
      },
      "required": [
      ]
    },
    "additionalProperties": {
      "type": "object",
      "properties": {
        "exclusionPatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "inclusionPatterns": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "crawlBotMessages": {
          "type": "boolean"
        },
        "excludeArchived": {
          "type": "boolean"
        },
        "conversationType": {
          "type": "array",
          "items": {
            "type": "string",
            "enum": [
              "PUBLIC_CHANNEL",
              "PRIVATE_CHANNEL",
              "GROUP_MESSAGE",
              "DIRECT_MESSAGE"
            ]
          }
        },
        "channelFilter": {
            "type": "object",
            "properties": {
              "private_channel": {
```

```
                "type": "array",
                "items": {
                  "type": "string"
                }
              },
              "public_channel": {
                "type": "array",
                "items": {
                  "type": "string"
                }
              }
          }
        },
        "channelIdFilter": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "sinceDate": {
          "anyOf": [
            {
              "type": "string",
              "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}Z$"
            },
            {
              "type": "string",
              "pattern": ""
            }
          ]
        },
        "lookBack": {
          "type": "string",
          "pattern": "^[0-9]*$"
        }
      },
      "required": [
      ]
    },
    "syncMode": {
      "type": "string",
      "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL",
```

```
          "CHANGE_LOG"
        ]
      },
      "type" : {
        "type" : "string",
        "pattern": "SLACK"
      },
      "enableIdentityCrawler": {
        "type": "boolean"
      },
      "secretArn": {
        "type": "string"
      }
    },
    "version": {
      "type": "string",
      "anyOf": [
        {
          "pattern": "1.0.0"
        }
      ]
    },
    "required": [
      "connectionConfiguration",
      "repositoryConfigurations",
      "syncMode",
      "additionalProperties",
      "secretArn",
      "type",
      "enableIdentityCrawler"
    ]
  }
```

Zendesk template schema

You include a JSON that contains the data source schema as part of TemplateConfiguration object. You provide the host URL as a part of the connection configuration or repository endpoint details. Also specify the type of data source as `ZENDESK`, a secret for your authentication credentials, and other necessary configurations. You then specify `TEMPLATE` as the `Type` when you call CreateDataSource.

You can use the template provided in this developer guide. See Zendesk JSON schema.

The following table describes the parameters of the Zendesk JSON schema.

Configuration	Description
connectionConfiguration	Configuration information for the endpoint for the data source.
repositoryEndpointMetadata	The endpoint information for the data source.
hostURL	The Zendesk host URL. For example, *https://yoursubdomain.zendesk.com*.
repositoryConfigurations	Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.
• ticket • ticketComment • ticketCommentAttachment • article • articleComment • articleAttachment • communityTopic • communityPostComment	A list of objects that map attributes or field names of Zendesk tickets to Amazon Kendra index field names. For more information, see [Mapping data source fields]().
secretARN	The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your Zendesk. The secret must contain a JSON structure with the following keys: host URL, client ID, client secret, user name, and password.
additionalProperties	Additional configuration options for your content in your data source

Configuration	Description
organizationNameFilter	You can choose to index tickets that exist within a specific **Organization**.
sinceDate	You can choose to configure a `sinceDate` parameter so that the Zendesk connector crawls content based on a specific `sinceDate` .
inclusionPatterns	A list of regular expression patterns to *include* certain files in your Zendesk data source. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence, and the file isn't included in the index.
exclusionPatterns	A list of regular expression patterns to *exclude* certain files in your Zendesk data source. Files that match the patterns are excluded from the index. Files that don't match the patterns are included in the index. If a file matches both an exclusion and inclusion pattern, the exclusion pattern takes precedence, and the file isn't included in the index.

Configuration	Description
• isCrawlTicket • isCrawlTicketComment • isCrawlTicketCommentAttachment • isCrawlArticle • isCrawlArticleComment • isCrawlArticleAttachment • isCrawlCommunityTopic • isCrawlCommunityPost • isCrawlCommunityPostComment	Input "`true`" to crawl these types of content.
type	Specify `ZENDESK` as your data source type.
useChangeLog	Input "`true`" to use the Zendesk change log to determine which documents require updating in the index. Depending on the change log's size, it might be faster to scan the documents in Zendesk. If you are syncing your Zendesk data source with your index for the first time, all documents are scanned.

Zendesk JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "hostUrl": {
              "type": "string",
              "pattern": "https:.*"
```

```
            }
          },
          "required": [
            "hostUrl"
          ]
        }
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "ticket": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": {
                "anyOf": [
                  {
                    "type": "object",
                    "properties": {
                      "indexFieldName": {
                        "type": "string"
                      },
                      "indexFieldType": {
                        "type": "string",
                        "enum": ["STRING", "STRING_LIST", "LONG", "DATE"]
                      },
                      "dataSourceFieldName": {
                        "type": "string"
                      },
                      "dateFieldFormat": {
                        "type": "string",
                        "pattern": "dd-MM-yyyy HH:mm:ss"
                      }
                    },
                    "required": [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
```

```
                ]
              }
            ]
          }
        }
      },
      "required": [
        "fieldMappings"
      ]
    },
    "ticketComment": {
      "type": "object",
      "properties": {
        "fieldMappings": {
          "type": "array",
          "items": {
            "anyOf": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
                    "enum": ["STRING", "STRING_LIST", "LONG", "DATE"]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  },
                  "dateFieldFormat": {
                    "type": "string",
                    "pattern": "dd-MM-yyyy HH:mm:ss"
                  }
                },
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"

                ]
              }
            ]
          }
```

```
            }
          },
          "required": [
            "fieldMappings"
          ]
        },
        "ticketCommentAttachment": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": {
                "anyOf": [
                  {
                    "type": "object",
                    "properties": {
                      "indexFieldName": {
                        "type": "string"
                      },
                      "indexFieldType": {
                        "type": "string",
                        "enum": ["STRING", "STRING_LIST", "LONG", "DATE"]
                      },
                      "dataSourceFieldName": {
                        "type": "string"
                      },
                      "dateFieldFormat": {
                        "type": "string",
                        "pattern": "dd-MM-yyyy HH:mm:ss"
                      }
                    },
                    "required": [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
                    ]
                  }
                ]
              }
            }
          },
          "required": [
            "fieldMappings"
          ]
```

```
        },
        "article": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": {
                "anyOf": [
                  {
                    "type": "object",
                    "properties": {
                      "indexFieldName": {
                        "type": "string"
                      },
                      "indexFieldType": {
                        "type": "string",
                        "enum": ["STRING", "STRING_LIST", "LONG", "DATE"]
                      },
                      "dataSourceFieldName": {
                        "type": "string"
                      },
                      "dateFieldFormat": {
                        "type": "string",
                        "pattern": "dd-MM-yyyy HH:mm:ss"
                      }
                    },
                    "required": [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
                    ]
                  }
                ]
              }
            }
          },
          "required": [
            "fieldMappings"
          ]
        },
        "communityPostComment": {
          "type": "object",
          "properties": {
            "fieldMappings": {
```

```
                "type": "array",
                "items": {
                  "anyOf": [
                    {
                      "type": "object",
                      "properties": {
                        "indexFieldName": {
                          "type": "string"
                        },
                        "indexFieldType": {
                          "type": "string",
                          "enum": ["STRING", "STRING_LIST", "LONG", "DATE"]
                        },
                        "dataSourceFieldName": {
                          "type": "string"
                        },
                        "dateFieldFormat": {
                          "type": "string",
                          "pattern": "dd-MM-yyyy HH:mm:ss"
                        }
                      },
                      "required": [
                        "indexFieldName",
                        "indexFieldType",
                        "dataSourceFieldName"
                      ]
                    }
                  ]
                }
              }
            },
            "required": [
              "fieldMappings"
            ]
          },
          "articleComment": {
            "type": "object",
            "properties": {
              "fieldMappings": {
                "type": "array",
                "items": {
                  "anyOf": [
                    {
                      "type": "object",
```

```
                    "properties": {
                      "indexFieldName": {
                        "type": "string"
                      },
                      "indexFieldType": {
                        "type": "string",
                        "enum": ["STRING", "STRING_LIST", "LONG", "DATE"]
                      },
                      "dataSourceFieldName": {
                        "type": "string"
                      },
                      "dateFieldFormat": {
                        "type": "string",
                        "pattern": "dd-MM-yyyy HH:mm:ss"
                      }
                    },
                    "required": [
                      "indexFieldName",
                      "indexFieldType",
                      "dataSourceFieldName"
                    ]
                  }
                ]
              }
            }
          },
          "required": [
            "fieldMappings"
          ]
        },
        "articleAttachment": {
          "type": "object",
          "properties": {
            "fieldMappings": {
              "type": "array",
              "items": {
                "anyOf": [
                  {
                    "type": "object",
                    "properties": {
                      "indexFieldName": {
                        "type": "string"
                      },
                      "indexFieldType": {
```

```
                    "type": "string",
                    "enum": ["STRING", "STRING_LIST", "LONG", "DATE"]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
                  },
                  "dateFieldFormat": {
                    "type": "string",
                    "pattern": "dd-MM-yyyy HH:mm:ss"
                  }
                },
                "required": [
                  "indexFieldName",
                  "indexFieldType",
                  "dataSourceFieldName"
                ]
              }
            ]
          }
        }
      },
      "required": [
        "fieldMappings"
      ]
    },
    "communityTopic": {
      "type": "object",
      "properties": {
        "fieldMappings": {
          "type": "array",
          "items": {
            "anyOf": [
              {
                "type": "object",
                "properties": {
                  "indexFieldName": {
                    "type": "string"
                  },
                  "indexFieldType": {
                    "type": "string",
                    "enum": ["STRING", "STRING_LIST", "LONG", "DATE"]
                  },
                  "dataSourceFieldName": {
                    "type": "string"
```

```
                        },
                        "dateFieldFormat": {
                          "type": "string",
                          "pattern": "dd-MM-yyyy HH:mm:ss"
                        }
                      },
                      "required": [
                        "indexFieldName",
                        "indexFieldType",
                        "dataSourceFieldName"
                      ]
                    }
                  ]
                }
              }
            },
            "required": [
              "fieldMappings"
            ]
          }
        }
      },
      "secretArn": {
        "type": "string",
        "minLength": 20,
        "maxLength": 2048
      },
      "additionalProperties": {
        "type": "object",
        "properties": {
          "organizationNameFilter": {
            "type": "array"
          },
          "sinceDate": {
            "type": "string",
            "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-9]{2}$"
          },
          "inclusionPatterns": {
            "type": "array"
          },
          "exclusionPatterns": {
            "type": "array"
          },
          "isCrawTicket": {
```

```
          "type": "string"
        },
        "isCrawTicketComment": {
          "type": "string"
        },
        "isCrawTicketCommentAttachment": {
          "type": "string"
        },
        "isCrawlArticle": {
          "type": "string"
        },
        "isCrawlArticleAttachment": {
          "type": "string"
        },
        "isCrawlArticleComment": {
          "type": "string"
        },
        "isCrawlCommunityTopic": {
          "type": "string"
        },
        "isCrawlCommunityPost": {
          "type": "string"
        },
        "isCrawlCommunityPostComment": {
          "type": "string"
        }
      }
    },
    "type": {
      "type": "string",
      "pattern": "ZENDESK"
    },
    "useChangeLog": {
      "type": "string",
      "enum": ["true", "false"]
    }
  },
  "version": {
    "type": "string",
    "anyOf": [
      {
        "pattern": "1.0.0"
      }
    ]
```

```
  },
  "additionalProperties": false,
  "required": [
    "connectionConfiguration",
    "repositoryConfigurations",
    "additionalProperties",
    "useChangeLog",
    "secretArn",
    "type"
  ]
}
```

Adobe Experience Manager

> **Note**
>
> Adobe Experience Manager connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

Adobe Experience Manager is a content management system that's used for creating website or mobile app content. You can use Amazon Kendra to connect to Adobe Experience Manager and index your pages and content assets.

Amazon Kendra supports Adobe Experience Manager (AEM) as a Cloud Service author instance and Adobe Experience Manager On-Premise author and publish instance.

You can connect Amazon Kendra to your Adobe Experience Manager data source using the Amazon Kendra console or the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Adobe Experience Manager data source connector, see Troubleshooting data sources.

Topics

- Supported features

- Prerequisites
- Connection instructions

Supported features

Adobe Experience Manager data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- OAuth 2.0 and basic authentication
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Adobe Experience Manager data source, make these changes in your Adobe Experience Manager and AWS accounts.

In Adobe Experience Manager, make sure you have:

- Access to an account with administrative privileges, or an admin user.
- Copied your Adobe Experience Manager host URL.

> **Note**
>
> (On-premise/server) Amazon Kendra checks if the endpoint information included in AWS Secrets Manager is the same the endpoint information specified in your data source configuration details. This helps protect against the confused deputy problem, which is a security issue where a user doesn't have permission to perform an action but uses Amazon Kendra as a proxy to access the configured secret and perform the action. If you later change your endpoint information, you must create a new secret to sync this information.

- Noted your basic authentication credentials of admin user name and password.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **Optional**: Configured OAuth 2.0 credentials in Adobe Experience Manager (AEM) as a Cloud Service or AEM On-Premise. If you use AEM On-Premise, the credentials include client ID, client secret, and private key. If you use AEM as a Cloud Service, the credentials include client ID, client secret, private key, organization ID, technical account ID, and Adobe Identity Management System (IMS) host. For more information about how to generate these credentials for AEM as a Cloud Service, see Adobe Experience Manager documentation. For AEM On-Premise, Adobe Granite OAuth 2.0 server implementation (com.adobe.granite.oauth.server) provides the support for OAuth 2.0 server functionalities in AEM.
- Checked each document is unique in Adobe Experience Manager and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Adobe Experience Manager authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Adobe Experience Manager data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Adobe Experience Manager data source, you must provide the necessary details of your Adobe Experience Manager data source so that Amazon Kendra can access your data. If you have not yet configured Adobe Experience Manager for Amazon Kendra, see [Prerequisites](#).

Console

To connect Amazon Kendra to Adobe Experience Manager

1. Sign in to the AWS Management Console and open the [Amazon Kendra console](#).
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Adobe Experience Manager connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Adobe Experience Manager connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

a. In **Name and description**, for **Data source name—**Enter a name for your data source. You can include hyphens but not spaces.

b. (Optional) **Description—**Enter an optional description for your data source.

c. In **Default language—**Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag—**Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

a. **Source—**Choose either **AEM On-Premise** or **AEM as a Cloud Service**.

Enter your Adobe Experience Manager host URL. For example, if you use AEM On-Premise, you include the hostname and port: *https://hostname:port*. Or, if you use AEM as a Cloud Service, you can use the author URL: *https://author-xxxxxx-xxxxxxx.adobeaemcloud.com*.

b. **SSL certificate location—**Enter the path to the SSL certificate stored in an Amazon S3 bucket. You use this to connect to AEM On-Premise with a secure SSL connection.

c. **Authorization—**Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

d. **Authentication—**Choose **Basic authentication** or **OAuth 2.0 authentication**. Then choose an existing AWS Secrets Manager secret or create a new secret to store your Adobe Experience Manager credentials. If you choose to create a new secret, an AWS Secrets Manager secret window opens.

If you chose **Basic authentication**, enter a name for the secret, the Adobe Experience Manager site user name and password. The user must have admin permission or be an admin user.

If you chose **OAuth 2.0 authentication** and you use AEM On-Premise, enter a name for the secret, client ID, client secret, and private key. If you use AEM as a Cloud

Service, enter a name for the secret, client ID, client secret, private key, organization ID, technical account ID, and Adobe Identity Management System (IMS) host.

Save an add your secret.

e. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

f. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

g. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

h. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. **Sync scope**—Set limits for crawling certain content types, page components, and roots paths, and filter content using regex expression patterns.

      i. **Content types**—Choose whether to crawl only pages or assets, or both.

      ii. (Optional) **Additional configuration**—Configure the following settings:

         - **Page components**—The specific names of page components. The Page Component is an extensible page component designed to work with the Adobe Experience Manager template editor and allows page header/footer and structure components to be assembled with the template editor.

- **Content fragment variations**—The specific names of content fragment variations. Content Fragments allow you to design, create, curate and publish page-independent content in Adobe Experience Manager. They allow you to prepare content ready for use in multiple locations/over multiple channels.
- **Root paths**—The root paths to specific content.
- **Regex patterns**—The regular expression patterns to include or exclude certain pages and assets.

b. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

c. **Time zone ID**—If you use AEM On-Premise and the time zone of your server is different than the time zone of the Amazon Kendra AEM connector or index, you can specify the server time zone to align with the AEM connector or index. The default time zone for AEM On-Premise is the time zone of the Amazon Kendra AEM connector or index. The default time zone for AEM as a Cloud Service is Greenwich Mean Time.

d. **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. Select from the Amazon Kendra generated default data source fields you want to map to your index. To add custom data source fields, create an index field name to map to and the field data type.

   b. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Adobe Experience Manager

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `AEM` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **AEM host URL**—Specify the Adobe Experience Manager host URL. For example, if you use AEM On-Premise, you include the hostname and port: *https://hostname:port*. Or, if you use AEM as a Cloud Service, you can use the author URL: *https://author-xxxxxx-xxxxxxx.adobeaemcloud.com*.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Authentication type**—Specify which type of authentication you want to use, either `Basic` or `OAuth2`.

- **AEM type**—Specify which type of Adobe Experience Manager you use, either CLOUD or ON_PREMISE.
- **Secret Amazon Resource Name (ARN)**—If you want to use basic authentication for either AEM On-Premise or Cloud, you provide a secret that stores your authentication credentials of your user name and password. You provide the Amazon Resource Name (ARN) of an AWS Secrets Manager secret. The secret is stored in a JSON structure with the following keys:

```
{
    "aemUrl": "Adobe Experience Manager On-Premise host URL",
    "username": "user name with admin permissions",
    "password": "password with admin permissions"
}
```

  If you want to use OAuth 2.0 authentication for AEM On-Premise, the secret is stored in a JSON structure with the following keys:

```
{
    "aemUrl": "Adobe Experience Manager host URL",
    "clientId": "client ID",
    "clientSecret": "client secret",
    "privateKey": "private key"
}
```

  If you want to use OAuth 2.0 authentication for AEM as a Cloud Service, the secret is stored in a JSON structure with the following keys:

```
{
    "clientId": "client ID",
    "clientSecret": "client secret",
    "privateKey": "private key",
    "orgId": "organization ID",
    "technicalAccountId": "technical account ID",
    "imsHost": "Adobe Identity Management System (IMS) host"
}
```

- **IAM role**—Specify RoleArn when you call CreateDataSource to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Adobe Experience Manager connector and Amazon Kendra. For more information, see IAM roles for Adobe Experience Manager data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Time zone ID**—If you use AEM On-Premise and the time zone of your server is different than the time zone of the Amazon Kendra AEM connector or index, you can specify the server time zone to align with the AEM connector or index.

  The default time zone for AEM On-Premise is the time zone of the Amazon Kendra AEM connector or index. The default time zone for AEM as a Cloud Service is Greenwich Mean Time.

  For information about the supported time zones IDs, see Adobe Experience Manager JSON schema.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain pages and assets.

  > **Note**
  >
  > Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.
- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **Field mappings**—Choose to map your Adobe Experience Manager data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Adobe Experience Manager template schema.

Alfresco

> **Note**
>
> Alfresco connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

Alfresco is a content management service that helps customers store and manage their content. You can use Amazon Kendra to index your Alfresco Document library, Wiki, and Blog.

Amazon Kendra supports Alfresco On-Premises and Alfresco Cloud (Platform as a Service).

You can connect Amazon Kendra to your Alfresco data source using the Amazon Kendra console or the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Alfresco data source connector, see Troubleshooting data sources.

Topics

- Connection instructions
- Learn more

Supported features

Amazon Kendra Alfresco data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- OAuth 2.0 and basic authentication
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Alfresco data source, make these changes in your Alfresco and AWS accounts.

In Alfresco, make sure you have:

- Copied your Alfresco repository URL and web application URL. If you only want to index a specific Alfresco site, then also copy the site ID.
- Noted your Alfresco authentication credentials, which include a user name and password with at least read permissions. If you want to use OAuth 2.0 authentication, you should add the user to the Alfresco administrators group.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **Optional**: Configured OAuth 2.0 credentials in Alfresco. The credentials include client ID, client secret, and token URL. For more information on how to configure clients for Alfresco On-

Premises, see Alfresco documentation. If you use Alfresco Cloud (PaaS), you must contact Hyland support for Alfresco OAuth 2.0 authentication.

- Checked each document is unique in Alfresco and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Alfresco authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Alfresco data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Alfresco data source, you must provide the necessary details of your Alfresco data source so that Amazon Kendra can access your data. If you have not yet configured Alfresco for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to Alfresco

1. Sign in to the AWS Management Console and open the [Amazon Kendra console](https://console.aws.amazon.com/kendra/).
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Alfresco connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Alfresco connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. **Alfresco type**—Choose whether you use Alfresco On-Premises/server or Alfresco Cloud (Platform as a Service).

   b. **Alfresco repository URL**—Enter your Alfresco repository URL. For example, if you use Alfresco Cloud (PaaS), the repository URL could be *https://company.alfrescocloud.com*. Or, if you use Alfresco On-Premises, the repository URL could be *https://company-alfresco-instance.company-domain.suffix:port*.

c. **Alfresco user application. URL**—Enter your Alfresco user interface URL. You can get the repository URL from your Alfresco administrator. For example, the user interface URL could be *https://example.com*.

d. **SSL certificate location**—Enter the path to the SSL certificate stored in an Amazon S3 bucket. You use this to connect to Alfresco On-Premises with a secure SSL connection.

e. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see [User context filtering].

f. **Authentication**—Choose **Basic authentication** or **OAuth 2.0 authentication**. Then choose an existing Secrets Manager secret or create a new secret to store your Alfresco credentials. If you choose to create a new secret, an AWS Secrets Manager secret window opens.

   If you chose **Basic authentication**, enter a name for the secret, the Alfresco user name, and password.

   If you chose **OAuth 2.0 authentication**, enter a name for the secret, client ID, client secret, and token URL.

g. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

h. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure [user context filtering] of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the [PutPrincipalMapping] API to upload user and group access information for user context filtering.

i. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

j. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. **Sync scope**—Set limits for crawling certain content and filter content using regex expression patterns.

   b. i. **Content**—Choose whether to crawl content marked with 'Aspects' in Alfresco, content within a specific Alfresco site, or content across all your Alfresco sites.

      ii. (Optional)**Additional configuration**—Set the following settings:

      - **Include comments**—Choose to include comments in Alfresco Document library and Blog.
      - **Regex patterns**—Regular expression patterns to include or exclude certain files.

   c. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

   - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
   - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

   d. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

   e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

a. Select from the Amazon Kendra generated default data source fields that you want to map to your index.

b. To add custom data source fields, create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Alfresco

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `ALFRESCO` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Alfresco site ID**—Specify the Alfresco site ID.
- **Alfresco repository URL**—Specify the Alfresco repository URL. You can get the repository URL from your Alfresco administrator. For example, if you use Alfresco Cloud (PaaS), the repository URL could be *https://company.alfrescocloud.com*. Or, if you use Alfresco On-Premises, the repository URL could be *https://company-alfresco-instance.company-domain.suffix:port*.
- **Alfresco web application URL**—Specify the Alfresco user interface URL. You can get the repository URL from your Alfresco administrator. For example, the user interface URL could be *https://example.com*.
- **Authentication type**—Specify which type of authentication you want to use, whether `OAuth2` or `Basic`.
- **Alfresco type**—Specify which type of Alfresco you use, whether `PAAS` (Cloud/Platform as a Service) or `ON_PREM` (On-Premises).
- **Secret Amazon Resource Name (ARN)**—If you want to use basic authentication, you provide a secret that stores your authentication credentials of your user name and password. You

provide the Amazon Resource Name (ARN) of an AWS Secrets Manager secret. The secret is stored in a JSON structure with the following keys:

```
{
    "username": "user name",
    "password": "password"
}
```

If you want to use OAuth 2.0 authentication, the secret is stored in a JSON structure with the following keys:

```
{
    "clientId": "client ID",
    "clientSecret": "client secret",
    "tokenUrl": "token URL"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Alfresco connector and Amazon Kendra. For more information, see [IAM roles for Alfresco data sources](#).

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see [Configuring Amazon Kendra to use an Amazon VPC](#).
- **Content type**—The type of content that you want to crawl, whether content marked with 'Aspects' in Alfresco, content within a specific Alfresco site, or content across all your Alfresco sites. You can also list specific 'Aspects' content.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain files.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the

inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **Field mappings**—Choose to map your Alfresco data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Alfresco template schema.

Learn more

To learn more about integrating Amazon Kendra with your Alfresco data source, see:

- Intelligently search Alfresco content using Amazon Kendra

Aurora (MySQL)

> **Note**
>
> Aurora (MySQL) connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

Aurora is a relational database management system (RDBMS) built for the cloud. If you are a Aurora user, you can use Amazon Kendra to index your Aurora (MySQL) data source. The Amazon Kendra Aurora (MySQL) data source connector supports Aurora MySQL 3 and Aurora Serverless MySQL 8.0.

You can connect Amazon Kendra to your Aurora (MySQL) data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Aurora (MySQL) data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Notes

Supported features

- Field mappings
- User context filtering
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Aurora (MySQL) data source, make these changes in your Aurora (MySQL) and AWS accounts.

In Aurora (MySQL), make sure you have:

- Noted your database user name and password.

> **Important**
>
> As a best practice, provide Amazon Kendra with read-only database credentials.

- Copied your database host url, port, and instance. You can find this information on the Amazon RDS console.
- Checked each document is unique in Aurora (MySQL) and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Aurora (MySQL) authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Aurora (MySQL) data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Aurora (MySQL) data source you must provide details of your Aurora (MySQL) credentials so that Amazon Kendra can access your data. If you have not yet configured Aurora (MySQL) for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Aurora (MySQL)

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Aurora (MySQL) connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Aurora (MySQL) connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. In **Source**, enter the following information:

   b. **Host** – Enter the database host URL, for example: `http://instance URL.region.rds.amazonaws.com`.

   c. **Port** – Enter the database port, for example, `5432`.

   d. **Instance**— Enter the database instance.

   e. In **Authentication**—enter the following information:

      - **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Aurora (MySQL) authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

        A. Enter following information in the **Create an AWS Secrets Manager secret window**:

I. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Aurora (MySQL)-' is automatically added to your secret name.

II. For **Database user name**, and **Password**—Enter the authentication credential values you copied from your database.

B. Choose **Save**.

f. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

g. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

h. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. In **Sync scope**, choose from the following options :

- **SQL query**—Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Kendra will crawl all database content that matches your query.
- **Primary key column**—Provide the primary key for the database table. This identifies a table within your database.
- **Title column**—Provide the name of the document title column within your database table.
- **Body column**—Provide the name of the document body column within your database table.

b. In **Additional configuration – *optional***, choose from the following options to sync specific content instead of syncing all files:

- **Change-detecting columns—**Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns.
- **Users' IDs column—**Enter the name of the column which contains User IDs to be allowed access to content.
- **Groups column—**Enter the name of the column that contains groups to be allowed access to content.
- **Source URLs column—**Enter the name of the column which contains Source URLs to be indexed.
- **Time stamps column—**Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
- **Time zones column—**Enter the name of the column which contains time zones for the content to be crawled.
- **Time stamps format—**Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

c. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

   - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
   - New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
   - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency—**How often Amazon Kendra will sync with your data source.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. Select from the generated default data source fields—**Document IDs**, **Document titles**, and **Source URLs**—you want to map to Amazon Kendra index.

   b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Aurora (MySQL)

You must specify the following using the [TemplateConfiguration](#) API:

- **Data source**—Specify the data source type as `JDBC` when you use the [TemplateConfiguration](#) JSON schema. Also specify the data source as `TEMPLATE` when you call the [CreateDataSource](#) API.
- **Database type**—You must specify the database type as `mySql`.
- **SQL query**—Specify SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source’s mechanism for tracking content changes and index content that changed since the last sync.

- CHANGE_LOG to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials you created in your Aurora (MySQL) account. The secret is stored in a JSON structure with the following keys:

```
{
    "user name": "database user name",
    "password": "password"
}
```

Note

We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify RoleArn when you call CreateDataSource to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Aurora (MySQL) connector and Amazon Kendra. For more information, see IAM roles for Aurora (MySQL) data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify VpcConfiguration when you call CreateDataSource. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include specific content using user IDs, groups, source URLs, time stamps, and time zones.
- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

- **Field mappings**—Choose to map your Aurora (MySQL) data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Aurora (MySQL) template schema.

Notes

- Deleted database rows will not be tracked in when Amazon Kendra checks for updated content.
- The size of field names and values in a row of your database can't exceed 400KB.
- If you have a large amount of data in your database data source, and do not want Amazon Kendra to index all your database content after the first sync, you can choose to sync only new, modified, or deleted documents.
- As a best practice, provide Amazon Kendra with read-only database credentials.
- As a best practice, avoid adding tables with sensitive data or personal identifiable information (PII).

Aurora (PostgreSQL)

> **Note**
>
> Aurora (PostgreSQL) connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

Aurora is a relational database management system (RDBMS) built for the cloud. If you are a Aurora user, you can use Amazon Kendra to index your Aurora (PostgreSQL) data source. The Amazon Kendra Aurora (PostgreSQL) data source connector supports Aurora PostgreSQL 1.

You can connect Amazon Kendra to your Aurora (PostgreSQL) data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Aurora (PostgreSQL) data source connector, see Troubleshooting data sources.

Topics

Supported features

- Field mappings
- User context filtering
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Aurora (PostgreSQL) data source, make these changes in your Aurora (PostgreSQL) and AWS accounts.

In Aurora (PostgreSQL), make sure you have:

- Noted your database user name and password.

> **Important**
>
> As a best practice, provide Amazon Kendra with read-only database credentials.

- Copied your database host url, port, and instance.
- Checked each document is unique in Aurora (PostgreSQL) and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Aurora (PostgreSQL) authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Aurora (PostgreSQL) data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Aurora (PostgreSQL) data source you must provide details of your Aurora (PostgreSQL) credentials so that Amazon Kendra can access your data. If you have not yet configured Aurora (PostgreSQL) for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Aurora (PostgreSQL)

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Aurora (PostgreSQL) connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Aurora (PostgreSQL) connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. In **Source**, enter the following information:

   b. **Host** – Enter the database host URL, for example: `http://`*`instance URL`*`.`*`region`*`.rds.amazonaws.com`.

   c. **Port** – Enter the database port, for example, `5432`.

   d. **Instance** – Enter the database instance, for example `postgres`.

e. **Enable SSL certificate location**—Choose to enter the Amazon S3 path to your SSL certificate file.

f. In **Authentication**—enter the following information:

   - **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Aurora (PostgreSQL) authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

     A. Enter following information in the **Create an AWS Secrets Manager secret window**:

        I. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Aurora (PostgreSQL)-' is automatically added to your secret name.

        II. For **Database user name**, and **Password**—Enter the authentication credential values you copied from your database.

     B. Choose **Save**.

g. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. In **Sync scope**, choose from the following options :

      - **SQL query**—Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Kendra will crawl all database content that matches your query.

- **Primary key column**—Provide the primary key for the database table. This identifies a table within your database.
- **Title column**—Provide the name of the document title column within your database table.
- **Body column**—Provide the name of the document body column within your database table.

b. In **Additional configuration – *optional***, choose from the following options to sync specific content instead of syncing all files:

- **Change-detecting columns**—Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns.
- **Users' IDs column**—Enter the name of the column which contains User IDs to be allowed access to content.
- **Groups column**—Enter the name of the column that contains groups to be allowed access to content.
- **Source URLs column**—Enter the name of the column which contains Source URLs to be indexed.
- **Time stamps column**—Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
- **Time zones column**—Enter the name of the column which contains time zones for the content to be crawled.
- **Time stamps format**—Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

c. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.

- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency**—How often Amazon Kendra will sync with your data source.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

    a. Select from the generated default data source fields—**Document IDs**, **Document titles**, and **Source URLs**—you want to map to Amazon Kendra index.

    b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

    c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Aurora (PostgreSQL)

You must specify the following using the TemplateConfiguration API:

- **Data source**—Specify the data source type as `JDBC` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Database type**—You must specify the database type as `postgresql`.
- **SQL query**—Specify SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.

- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials you created in your Aurora (PostgreSQL) account. The secret is stored in a JSON structure with the following keys:

```
{
    "user name": "database user name",
    "password": "password"
}
```

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Aurora (PostgreSQL) connector and Amazon Kendra. For more information, see IAM roles for Aurora (PostgreSQL) data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include specific content using user IDs, groups, source URLs, time stamps, and time zones.
- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Aurora (PostgreSQL) data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Aurora (PostgreSQL) template schema.

Notes

- Deleted database rows will not be tracked in when Amazon Kendra checks for updated content.
- The size of field names and values in a row of your database can't exceed 400KB.
- If you have a large amount of data in your database data source, and do not want Amazon Kendra to index all your database content after the first sync, you can choose to sync only new, modified, or deleted documents.
- As a best practice, provide Amazon Kendra with read-only database credentials.
- As a best practice, avoid adding tables with sensitive data or personal identifiable information (PII).

Amazon FSx (Windows)

Amazon FSx (Windows) is a fully managed, cloud based file server system that offers shared storage capabilities. If you're an Amazon FSx (Windows) user, you can use Amazon Kendra to index your Amazon FSx (Windows) data source.

> **Note**
>
> Amazon Kendra now supports an upgraded Amazon FSx (Windows) connector.
> The console has been automatically upgraded for you. Any new connectors you create on the console will use the upgraded architecture. If you use the API, you must now use the TemplateConfiguration object instead of the `FSxConfiguration` object to configure your connector.
> Connectors configured using the older console and API architecture will continue to function as configured. However, you won't be able to edit or update them. If you want to edit or update your connector configuration, you must create a new connector.
> We recommended migrating your connector workflow to the upgraded version. Support for connectors configured using the older architecture is scheduled to end by June 2024.

You can connect Amazon Kendra to your Amazon FSx (Windows) data source using the Amazon Kendra console, or the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Amazon FSx (Windows) data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more

Supported features

Amazon Kendra Amazon FSx (Windows) data source connector supports the following features:

- Field mappings

- User access control
- User identity crawling
- Inclusion and exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Amazon FSx (Windows) data source, check the details of your Amazon FSx (Windows) and AWS accounts.

For Amazon FSx (Windows), make sure you have:

- Set up Amazon FSx (Windows) with read and mounting permissions.
- Noted your file system ID. You can find your file system ID on the File Systems dashboard in the Amazon FSx (Windows) console.
- Configured a virtual private cloud using Amazon VPC where your Amazon FSx (Windows) file system resides.
- Noted your Amazon FSx (Windows) authentication credentials for an Active Directory user account. This includes your Active Directory user name with your DNS domain name (for example, *user@corp.example.com*) and password.

> **Note**
> Use only the necessary credentials required for the connector to function. Do not use privileged credentials like domain admin.

> **Note**
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Checked each document is unique in Amazon FSx (Windows) and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Amazon FSx (Windows) authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Amazon FSx (Windows) data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Amazon FSx (Windows) data source, you must provide the necessary details of your Amazon FSx (Windows) data source so that Amazon Kendra can access your data. If you have not yet configured Amazon FSx (Windows) for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to your Amazon FSx (Windows) file system

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Amazon FSx (Windows) connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Amazon FSx (Windows) connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.
   b. (Optional) **Description**—Enter an optional description for your data source.
   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.
   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.
   e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. **Amazon FSx (Windows) file system ID**—Select from the dropdown your existing file system ID, fetched from Amazon FSx (Windows). Or, create an Amazon FSx (Windows) file system. You can find your file system ID on the File Systems dashboard in the Amazon FSx (Windows) console.
   b. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to

filter search results based on the user or their group access to documents. For more information, see [User context filtering](#).

c. **Authentication**—Choose an existing AWS Secrets Manager secret, or create a new secret to store your file system credentials. If you choose to create a new secret, an AWS Secrets Manager secret window opens.

   Provide a secret that stores your authentication credentials of your user name and password. The user name must include your DNS domain name. For example, *user@corp.example.com*.

   Save and add your secret.

d. **Virtual Private Cloud (VPC)**—You must select an Amazon VPC where your Amazon FSx (Windows) resides. You include the VPC subnet and security groups. See [Configuring an Amazon VPC](#).

e. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

   > **Note**
   >
   > IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

f. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. **Sync scope, Regex patterns**—Add regular expression patterns to include or exclude certain files.

   b. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

      - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.

- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

c. **Sync run schedule**—For **Frequency**, choose how often to sync your data source content and update your index.

d. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. Select from the Amazon Kendra generated default fields of your files that you want to map to your index. To add custom data source fields, create an index field name to map to and the field data type.

   b. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to your Amazon FSx (Windows) file system

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `FSX` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **File system ID**—The identifier of the Amazon FSx (Windows) file system. You can find your file system ID on the File Systems dashboard in the Amazon FSx (Windows) console.
- **File system type**—Specify the type of file system as `WINDOWS`.
- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.

> **Note**
>
> You must select an Amazon VPC where your Amazon FSx (Windows) resides. You include the VPC subnet and security groups.

- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Amazon FSx (Windows) account. The secret is stored in a JSON structure with the following keys:

```
{
    "username": "user@corp.example.com",
    "password": "password"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the

Amazon FSx (Windows) connector and Amazon Kendra. For more information, see IAM roles for Amazon FSx (Windows) data sources.

You can also add the following optional features:

- **Inclusion and exclusion filters**—Specify whether to include or exclude certain files.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Access control list (ACL)**—Specify whether to crawl ACL information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

> **Note**
>
> To test user context filtering on a user, you must include the DNS domain name as part of the user name when you issue the query. You must have administrative permissions of the Active Directory domain. You can also test user context filtering on a group name.

- **Field mappings**—Choose to map your Amazon FSx (Windows) data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see [Amazon FSx (Windows) template schema](#).

Learn more

To learn more about integrating Amazon Kendra with your Amazon FSx (Windows) data source, see:

- [Securely search unstructured data on Windows file systems with the Amazon Kendra connector for Amazon FSx (Windows) for Windows File Server](#).

Amazon FSx (NetApp ONTAP)

Amazon FSx (NetApp ONTAP) is a fully managed, cloud based file server system that offers shared storage capabilities. If you're an Amazon FSx (NetApp ONTAP) user, you can use Amazon Kendra to index your Amazon FSx (NetApp ONTAP) data source.

You can connect Amazon Kendra to your Amazon FSx (NetApp ONTAP) data source using the [Amazon Kendra console](#), or the [TemplateConfiguration](#) API.

For troubleshooting your Amazon Kendra Amazon FSx (NetApp ONTAP) data source connector, see [Troubleshooting data sources](#).

Topics

Supported features

Amazon Kendra Amazon FSx (NetApp ONTAP) data source connector supports the following features:

- Field mappings
- User access control
- Inclusion and exclusion filters

- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Amazon FSx (NetApp ONTAP) data source, check the details of your Amazon FSx (NetApp ONTAP) and AWS accounts.

For Amazon FSx (NetApp ONTAP), make sure you have:

- Set up Amazon FSx (NetApp ONTAP) with read and mounting permissions.
- Noted your file system ID. You can find your file system ID on the File Systems dashboard in the Amazon FSx (NetApp ONTAP) console.
- Noted the storage virtual machine (SVM) ID used with your file system. You can find your SVM ID by going to the File Systems dashboard in the Amazon FSx (NetApp ONTAP) console, selecting your file system ID, and then selecting **Storage virtual machines**.
- Configured a virtual private cloud using Amazon VPC where your Amazon FSx (NetApp ONTAP) file system resides.
- Noted your Amazon FSx (NetApp ONTAP) authentication credentials for an Active Directory user account. This includes your Active Directory user name with your DNS domain name (for example, *user@corp.example.com*) and password. If you use the Network File System (NFS) protocol for your Amazon FSx (NetApp ONTAP) file system, the authentication credentials include a left ID, right ID, and pre-shared key.

> **Note**
>
> Use only the necessary credentials required for the connector to function. Do not use privileged credentials like domain admin.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Checked each document is unique in Amazon FSx (NetApp ONTAP) and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Amazon FSx (NetApp ONTAP) authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Amazon FSx (NetApp ONTAP) data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Amazon FSx (NetApp ONTAP) data source, you must provide the necessary details of your Amazon FSx (NetApp ONTAP) data source so that Amazon Kendra can access your data. If you have not yet configured Amazon FSx (NetApp ONTAP) for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to your Amazon FSx (NetApp ONTAP) file system

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Amazon FSx (NetApp ONTAP) connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Amazon FSx (NetApp ONTAP) connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

    a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

    b. (Optional) **Description**—Enter an optional description for your data source.

    c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

    d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

    e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

    a. **Source**—Provide your file system information.

    - **File system protocol**—Choose the protocol of your Amazon FSx (NetApp ONTAP) file system. You can choose either Common Internet File System (CIFS) protocol, or the Network File System (NFS) protocol for Linux.
    - **Amazon FSx (NetApp ONTAP) file system ID**—Select from the dropdown your existing file system ID, fetched from Amazon FSx (NetApp ONTAP). Or, create an

Amazon FSx (NetApp ONTAP) file system. You can find your file system ID on the File Systems dashboard in the Amazon FSx (NetApp ONTAP) console.

- **SVM ID** (Amazon FSx (NetApp ONTAP) for NetApp ONTAP only)—Provide the storage virtual machine (SVM) ID of your Amazon FSx (NetApp ONTAP) NetApp ONTAP. You can find your SVM ID by going to the File Systems dashboard in the Amazon FSx (NetApp ONTAP) console, selecting your file system ID, and selecting **Storage virtual machines**.

b. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

c. **Authentication**—Choose an existing AWS Secrets Manager secret, or create a new secret to store your file system credentials. If you choose to create a new secret, an AWS Secrets Manager secret window opens.

   Provide a secret that stores your authentication credentials of your user name and password. The user name must include your DNS domain name. For example, *user@corp.example.com*.

   If you use the NFS protocol for your Amazon FSx (NetApp ONTAP) file system, provide a secret that stores your authentication credentials of left ID, right ID, and pre-shared key.

   Save and add your secret.

d. **Virtual Private Cloud (VPC)**—You must select an Amazon VPC where your Amazon FSx (NetApp ONTAP) resides. You include the VPC subnet and security groups. See Configuring an Amazon VPC.

e. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

f. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. **Sync scope, Regex patterns**—Add regular expression patterns to include or exclude certain files.

   b. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

      - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
      - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

   c. **Sync run schedule**—For **Frequency**, choose how often to sync your data source content and update your index.

   d. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. Select from the Amazon Kendra generated default fields of your files that you want to map to your index. To add custom data source fields, create an index field name to map to and the field data type.

   b. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to your Amazon FSx (NetApp ONTAP) file system

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `FSXONTAP` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **File system ID**—The identifier of the Amazon FSx (NetApp ONTAP) file system. You can find your file system ID on the File Systems dashboard in the Amazon FSx (NetApp ONTAP) console.
- **SVM ID**—The storage virtual machine (SVM) ID used with your file system. You can find your SVM ID by going to the File Systems dashboard in the Amazon FSx (NetApp ONTAP) console, selecting your file system ID, and then selecting **Storage virtual machines**.
- **Protocol type**—Specify whether you use the Common Internet File System (CIFS) protocol, or the Network File System (NFS) protocol for Linux.
- **File system type**—Specify the type of file system as either `FSXONTAP`.
- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.

  > **Note**
  >
  > You must select an Amazon VPC where your Amazon FSx (NetApp ONTAP) resides. You include the VPC subnet and security groups.

- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Amazon FSx (NetApp ONTAP) account. The secret is stored in a JSON structure with the following keys:

  ```
  {
      "username": "user@corp.example.com",
      "password": "password"
  }
  ```

  If you use the NFS protocol for your Amazon FSx (NetApp ONTAP) file system, the secret is stored in a JSON structure with the following keys:

  ```
  {
  ```

```
    "leftId": "left ID",
    "rightId": "right ID",
    "preSharedKey": "pre-shared key"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Amazon FSx (NetApp ONTAP) connector and Amazon Kendra. For more information, see IAM roles for Amazon FSx (NetApp ONTAP) data sources.

You can also add the following optional features:

- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain files.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Access control list (ACL)**—Specify whether to crawl ACL information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

> **Note**
>
> To test user context filtering on a user, you must include the DNS domain name as part of the user name when you issue the query. You must have administrative permissions of the Active Directory domain. You can also test user context filtering on a group name.

- **Field mappings**—Choose to map your Amazon FSx (NetApp ONTAP) data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Amazon FSx (NetApp ONTAP) template schema.

Amazon RDS/Aurora

You can index documents that are stored in a database using a database data source. After you provided connection information for the database, Amazon Kendra connects and indexes documents.

Amazon Kendra supports the following databases:

- Amazon Aurora MySQL
- Amazon Aurora PostgreSQL
- Amazon RDS for MySQL
- Amazon RDS for PostgreSQL

Note

Serverless Aurora databases are not supported.

Important

This Amazon RDS/Aurora connector is scheduled for deprecation by the end of 2023. Amazon Kendra now supports new database data source connectors. For an improved experience, we recommend you choose from the following new connectors for your use case:

- Aurora (MySQL)
- Aurora (PostgreSQL)
- Amazon RDS (MySQL)
- Amazon RDS (Microsoft SQL Server)
- Amazon RDS (Oracle)
- Amazon RDS (PostgreSQL)
- IBM DB2
- Microsoft SQL Server
- MySQL
- Oracle Database
- PostgreSQL

You can connect Amazon Kendra to your database data source using the Amazon Kendra console and the DatabaseConfiguration API.

For troubleshooting your Amazon Kendra database data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions

Supported features

Amazon Kendra database data source connector supports the following features:

- Field mappings
- User context filtering
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your database data source, make these changes in your database and AWS accounts.

In your database, make sure you have:

- Noted your basic authentication credentials of user name and password for your database.
- Copied the host name, port number, host address, the name of the database, and the name of the data table that contains the document data. For PostgreSQL, the data table must be a public table or public schema.

> **Note**
>
> The host and port tell Amazon Kendra where to find the database server on the internet. The database name and table name tell Amazon Kendra where to find the document data on the database server.

- Copied the names of the columns in the data table that contain the document data. You must include the document ID, document body, columns to detect if a document has changed (for example, last updated column), and optional data table columns that map to custom index fields. You can also map any of the Amazon Kendra reserved field names to a table column.
- Copied the database engine type information such as whether you use Amazon RDS for MySQL or another type.
- Checked each document is unique in database and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your database authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your database data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your database data source, you must provide the necessary details of your database data source so that Amazon Kendra can access your data. If you have not yet configured database for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to a database

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **database connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **database connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

    a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

    b. (Optional) **Description**—Enter an optional description for your data source.

    c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

    d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

    e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

    a. **Endpoint**—A DNS host name, an IPv4 address, or an IPv6 address.

    b. **Port**—A port number.

    c. **Database**—Database name.

    d. **Table name**—Table name.

    e. For **Type of authentication**, choose between **Existing** and **New** to store your database authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

    - Enter following information in the **Create an AWS Secrets Manager secret window**:

        A. **Secret name**—A name for your secret. The prefix 'AmazonKendra-database-' is automatically added to your secret name.

   B. For **User name** and **Password**—Enter the authentication credential values from your database account.

   C. Choose **Save authentication**.

f. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

> **Note**
>
> You must use a private subnet. If your RDS instance is in a public subnet in your VPC, you can create a private subnet that has outbound access to a NAT gateway in the public subnet. The subnets provided in the VPC configuration must be in either US West (Oregon), US East (N. Virginia), EU (Ireland).

g. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

h. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. Select between **Aurora MySQL**, **MySQL**, **Aurora PostgreSQL**, and **PostgreSQL** based on your use case.

   b. **Enclose SQL identifiers with double quotes**—Select to enclose SQL identifiers in double quotes. For example, “columnName”.

   c. **ACL column** and **Change detecting columns**—Configure the columns that Amazon Kendra uses for change detection (for example, last updated column) and your access control list.

   d. In **Sync run schedule**, for **Frequency**—Choose how often Amazon Kendra will sync with your data source.

   e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

a. **Amazon Kendra default field mappings**—Select from the Amazon Kendra generated default data source fields you want to map to your index. You must add the **Database column** values for `document_id` and `document_body`

b. **Custom field mappings**—To add custom data source fields to create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to a database

You must specify the following the DatabaseConfiguration API:

- **ColumnConfiguration**—Information about where the index should get the document information from the database. For more details, see ColumnConfiguration. You must specify the `DocumentDataColumnName` (document body or main text) and `DocumentIdColumnName`, and `ChangeDetectingColumn` (for example, last updated column) fields. The column mapped to the `DocumentIdColumnName` field must be an integer column. The following example shows a simple column configuration for a database data source:

```
"ColumnConfiguration": {
    "ChangeDetectingColumns": [
        "LastUpdateDate",
        "LastUpdateTime"
    ],
    "DocumentDataColumnName": "TextColumn",
    "DocumentIdColumnName": "IdentifierColumn",
    "DocoumentTitleColumnName": "TitleColumn",
    "FieldMappings": [
        {
            "DataSourceFieldName": "AbstractColumn",
            "IndexFieldName": "Abstract"
        }
```

```
    ]
}
```

- **ConnectionConfiguration**—Configuration information that's required to connect to a database. For more details, see ConnectionConfiguration.
- **DatabaseEngineType**—The type of database engine that runs the database. The `DatabaseHost` field for `ConnectionConfiguration` must be the Amazon Relational Database Service (Amazon RDS) instance endpoint for the database. Don't use the cluster endpoint.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your database account. The secret is stored in a JSON structure with the following keys:

```
{
    "username": "user name",
    "password": "password"
}
```

The following example shows a database configuration, including the secret ARN.

```

"DatabaseConfiguration": {
"ConnectionConfiguration": {
"DatabaseHost": "host.subdomain.domain.tld",
        "DatabaseName": "DocumentDatabase",
        "DatabasePort": 3306,
        "SecretArn": "arn:aws:secretmanager:region:account ID:secret/secret name",
        "TableName": "DocumentTable"
    }
}
```

Note

We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the database connector and Amazon Kendra. For more information, see IAM roles for database data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` as part of the data source configuration. See Configuring Amazon Kendra to use a VPC.

> **Note**
>
> You must only use a private subnet. If your RDS instance is in a public subnet in your VPC, you can create a private subnet that has outbound access to a NAT gateway in the public subnet. The subnets provided in the VPC configuration must be in either US West (Oregon), US East (N. Virginia), EU (Ireland).

- **Field mappings**—Choose to map your database data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

Amazon RDS (Microsoft SQL Server)

> **Note**
>
> Amazon RDS (Microsoft SQL Server) connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

SQL Server is database management system developed by Microsoft. Amazon RDS for SQL Server makes it easy to set up, operate, and scale SQL Server deployments in the cloud. If you are a Amazon RDS (Microsoft SQL Server) user, you can use Amazon Kendra to index your Amazon RDS (Microsoft SQL Server) data source. The Amazon Kendra JDBC data source connector supports Microsoft SQL Server 2019.

You can connect Amazon Kendra to your Amazon RDS (Microsoft SQL Server) data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Amazon RDS (Microsoft SQL Server) data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Notes

Supported features

- Field mappings
- User context filtering
- Inclusion/exclusion filters

- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Amazon RDS (Microsoft SQL Server) data source, make these changes in your Amazon RDS (Microsoft SQL Server) and AWS accounts.

In Amazon RDS (Microsoft SQL Server), make sure you have:

- Noted your database user name and password.

> **Important**
>
> As a best practice, provide Amazon Kendra with read-only database credentials.

- Copied your database host url, port, and instance.
- Checked each document is unique in Amazon RDS (Microsoft SQL Server) and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Amazon RDS (Microsoft SQL Server) authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Amazon RDS (Microsoft SQL Server) data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Amazon RDS (Microsoft SQL Server) data source you must provide details of your Amazon RDS (Microsoft SQL Server) credentials so that Amazon Kendra can access your data. If you have not yet configured Amazon RDS (Microsoft SQL Server) for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Amazon RDS (Microsoft SQL Server)

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Amazon RDS (Microsoft SQL Server) connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Amazon RDS (Microsoft SQL Server) connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

a. In **Name and description**, for **Data source name—**Enter a name for your data source. You can include hyphens but not spaces.

b. (Optional) **Description—**Enter an optional description for your data source.

c. In **Default language—**Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag—**Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. In **Source**, enter the following information:

   b. **Host—** Enter the database host name.

   c. **Port—** Enter the database port.

   d. **Instance—** Enter the database instance.

   e. **Enable SSL certificate location—**Choose to enter the Amazon S3 path to your SSL certificate file.

   f. In **Authentication—**enter the following information:

      - **AWS Secrets Manager secret—**Choose an existing secret or create a new Secrets Manager secret to store your Amazon RDS (Microsoft SQL Server) authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

        A. Enter following information in the **Create an AWS Secrets Manager secret window**:

           I. **Secret name—**A name for your secret. The prefix ‘AmazonKendra-Amazon RDS (Microsoft SQL Server)-’ is automatically added to your secret name.

           II. For **Database user name**, and **Password—**Enter the authentication credential values you copied from your database.

        B. Choose **Save**.

   g. **Virtual Private Cloud (VPC)—**You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. In **Sync scope**, choose from the following options :

   - **SQL query**—Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.

     > **Note**
     >
     > If a table name includes special characters (non alphanumeric) in the name, you must use square brackets around the table name. For example, `select * from [my-database-table]`

   - **Primary key column**—Provide the primary key for the database table. This identifies a table within your database.
   - **Title column**—Provide the name of the document title column within your database table.
   - **Body column**—Provide the name of the document body column within your database table.

   b. In **Additional configuration – *optional***, choose from the following options to sync specific content instead of syncing all files:

   - **Change-detecting columns**—Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns.

- **User IDs column—**Enter the name of the column which contains User IDs to be allowed access to content.
- **Groups column—**Enter the name of the column that contains groups to be allowed access to content.
- **Source URLs column—**Enter the name of the column which contains Source URLs to be indexed.
- **Time stamps column—**Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
- **Time zones column—**Enter the name of the column which contains time zones for the content to be crawled.
- **Time stamps format**—Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

c. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency—**How often Amazon Kendra will sync with your data source.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. Select from the generated default data source fields**—Document IDs**, **Document titles**, and **Source URLs**—you want to map to Amazon Kendra index.

b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Amazon RDS (Microsoft SQL Server)

You must specify the following using the TemplateConfiguration API:

- **Data source**—Specify the data source type as `JDBC` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Database type**—You must specify the database type as `sqlserver`.
- **SQL query**—Specify SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.

> **Note**
>
> If a table name includes special characters (non alphanumeric) in the name, you must use square brackets around the table name. For example, *`select * from [my-database-table]`*

- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.

    - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
    - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials you created in your Amazon RDS (Microsoft SQL Server) account. The secret is stored in a JSON structure with the following keys:

```
{
    "user name": "database user name",
    "password": "password"
}
```

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Amazon RDS (Microsoft SQL Server) connector and Amazon Kendra. For more information, see [IAM roles for Amazon RDS (Microsoft SQL Server) data sources](#).

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see [Configuring Amazon Kendra to use an Amazon VPC](#).
- **Inclusion and exclusion filters**—You can specify whether to include specific content using user IDs, groups, source URLs, time stamps, and time zones.

- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Amazon RDS (Microsoft SQL Server) data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Amazon RDS (Microsoft SQL Server) template schema.

Notes

- Deleted database rows will not be tracked in when Amazon Kendra checks for updated content.
- The size of field names and values in a row of your database can't exceed 400KB.
- If you have a large amount of data in your database data source, and do not want Amazon Kendra to index all your database content after the first sync, you can choose to sync only new, modified, or deleted documents.
- As a best practice, provide Amazon Kendra with read-only database credentials.
- As a best practice, avoid adding tables with sensitive data or personal identifiable information (PII).

Amazon RDS (MySQL)

> **Note**
>
> Amazon RDS (MySQL) connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users

can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

Amazon RDS (Amazon Relational Database Service) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. If you are a Amazon RDS user, you can use Amazon Kendra to index your Amazon RDS (MySQL) data source. The Amazon Kendra data source connector supports Amazon RDS MySql 5.6, 5.7, and 8.0.

You can connect Amazon Kendra to your Amazon RDS (MySQL) data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Amazon RDS (MySQL) data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Notes

Supported features

- Field mappings
- User context filtering
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Amazon RDS (MySQL) data source, make these changes in your Amazon RDS (MySQL) and AWS accounts.

In Amazon RDS (MySQL), make sure you have:

- Noted your database user name and password.

> **Important**
>
> As a best practice, provide Amazon Kendra with read-only database credentials.

- Copied your database host url, port, and instance. You can find this information on the Amazon RDS console.
- Checked each document is unique in Amazon RDS (MySQL) and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Amazon RDS (MySQL) authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don’t have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Amazon RDS (MySQL) data source to Amazon

Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Amazon RDS (MySQL) data source you must provide details of your Amazon RDS (MySQL) credentials so that Amazon Kendra can access your data. If you have not yet configured Amazon RDS (MySQL) for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Amazon RDS (MySQL)

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Amazon RDS (MySQL) connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Amazon RDS (MySQL) connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.
   b. (Optional) **Description**—Enter an optional description for your data source.
   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.
   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.
   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

a. In **Source**, enter the following information:

b. **Host** – Enter the database host URL, for example: `http://`*`instance URL`*`.`*`region`*`.rds.amazonaws.com`.

c. **Port** – Enter the database port, for example, `5432`.

d. **Instance** – Enter the database instance, for example `postgres`.

e. **Enable SSL certificate location**—Choose to enter the Amazon S3 path to your SSL certificate file.

f. In **Authentication**—enter the following information:

- **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Amazon RDS (MySQL) authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

  A. Enter following information in the **Create an AWS Secrets Manager secret window**:

     I. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Amazon RDS (MySQL)-' is automatically added to your secret name.

     II. For **Database user name**, and **Password**—Enter the authentication credential values you copied from your database.

  B. Choose **Save**.

g. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. In **Sync scope**, choose from the following options :

- **SQL query—**Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Kendra will crawl all database content that matches your query.
- **Primary key column—**Provide the primary key for the database table. This identifies a table within your database.
- **Title column—**Provide the name of the document title column within your database table.
- **Body column—**Provide the name of the document body column within your database table.

b. In **Additional configuration – *optional***, choose from the following options to sync specific content instead of syncing all files:

- **Change-detecting columns—**Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns.
- **Users' IDs column—**Enter the name of the column which contains User IDs to be allowed access to content.
- **Groups column—**Enter the name of the column that contains groups to be allowed access to content.
- **Source URLs column—**Enter the name of the column which contains Source URLs to be indexed.
- **Time stamps column—**Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
- **Time zones column—**Enter the name of the column which contains time zones for the content to be crawled.
- **Time stamps format—**Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

c. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency**—How often Amazon Kendra will sync with your data source.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

a. Select from the generated default data source fields—**Document IDs**, **Document titles**, and **Source URLs**—you want to map to Amazon Kendra index.

b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Amazon RDS (MySQL)

You must specify the following using the TemplateConfiguration API:

- **Data source**—Specify the data source type as `JDBC` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Database type**—You must specify the database type as `mySql`.

- **SQL query**—Specify SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials you created in your Amazon RDS (MySQL) account. The secret is stored in a JSON structure with the following keys:

  ```
  {
      "user name": "database user name",
      "password": "password"
  }
  ```

  > **Note**
  >
  > We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the

Amazon RDS (MySQL) connector and Amazon Kendra. For more information, see IAM roles for Amazon RDS (MySQL) data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include specific content using user IDs, groups, source URLs, time stamps, and time zones.
- **Field mappings**—Choose to map your Amazon RDS (MySQL) data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

For a list of other important JSON keys to configure, see Amazon RDS (MySQL) template schema.

Notes

- Deleted database rows will not be tracked in when Amazon Kendra checks for updated content.
- The size of field names and values in a row of your database can't exceed 400KB.
- If you have a large amount of data in your database data source, and do not want Amazon Kendra to index all your database content after the first sync, you can choose to sync only new, modified, or deleted documents.

- As a best practice, provide Amazon Kendra with read-only database credentials.
- As a best practice, avoid adding tables with sensitive data or personal identifiable information (PII).

Amazon RDS (Oracle)

> **Note**
>
> Amazon RDS (Oracle) connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

Amazon RDS (Amazon Relational Database Service) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. If you are a Amazon RDS (Oracle) user, you can use Amazon Kendra to index your Amazon RDS (Oracle) data source. The Amazon Kendra Amazon RDS (Oracle) data source connector supports Amazon RDS Oracle Database 21c, Oracle Database 19c, Oracle Database 12c.

You can connect Amazon Kendra to your Amazon RDS (Oracle) data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Amazon RDS (Oracle) data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Notes

Supported features

- Field mappings

- User context filtering
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Amazon RDS (Oracle) data source, make these changes in your Amazon RDS (Oracle) and AWS accounts.

In Amazon RDS (Oracle), make sure you have:

- Noted your database user name and password.

> **Important**
>
> As a best practice, provide Amazon Kendra with read-only database credentials.

- Copied your database host url, port, and instance.
- Checked each document is unique in Amazon RDS (Oracle) and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Amazon RDS (Oracle) authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Amazon RDS (Oracle) data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Amazon RDS (Oracle) data source you must provide details of your Amazon RDS (Oracle) credentials so that Amazon Kendra can access your data. If you have not yet configured Amazon RDS (Oracle) for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Amazon RDS (Oracle)

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Amazon RDS (Oracle) connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Amazon RDS (Oracle) connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

a. In **Name and description**, for **Data source name—**Enter a name for your data source. You can include hyphens but not spaces.

b. (Optional) **Description—**Enter an optional description for your data source.

c. In **Default language—**Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag—**Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. In **Source**, enter the following information:

   b. **Host—** Enter the database host name.

   c. **Port—** Enter the database port.

   d. **Instance—** Enter the database instance.

   e. **Enable SSL certificate location—**Choose to enter the Amazon S3 path to your SSL certificate file.

   f. In **Authentication—**enter the following information:

   - **AWS Secrets Manager secret—**Choose an existing secret or create a new Secrets Manager secret to store your Amazon RDS (Oracle) authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

     A. Enter following information in the **Create an AWS Secrets Manager secret window**:

        I. **Secret name—**A name for your secret. The prefix ‘AmazonKendra-Amazon RDS (Oracle)-’ is automatically added to your secret name.

        II. For **Database user name**, and **Password—**Enter the authentication credential values you copied from your database.

     B. Choose **Save**.

   g. **Virtual Private Cloud (VPC)—**You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. In **Sync scope**, choose from the following options :

- **SQL query**—Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
- **Primary key column**—Provide the primary key for the database table. This identifies a table within your database.
- **Title column**—Provide the name of the document title column within your database table.
- **Body column**—Provide the name of the document body column within your database table.

b. In **Additional configuration – *optional***, choose from the following options to sync specific content instead of syncing all files:

- **Change-detecting columns**—Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns.
- **Users' IDs column**—Enter the name of the column which contains User IDs to be allowed access to content.
- **Groups column**—Enter the name of the column that contains groups to be allowed access to content.
- **Source URLs column**—Enter the name of the column which contains Source URLs to be indexed.

- **Time stamps column—**Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
- **Time zones column—**Enter the name of the column which contains time zones for the content to be crawled.
- **Time stamps format—**Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

c. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency—**How often Amazon Kendra will sync with your data source.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

a. Select from the generated default data source fields—**Document IDs**, **Document titles**, and **Source URLs—**you want to map to Amazon Kendra index.

b. **Add field—**To add custom data source fields to create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this

page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Amazon RDS (Oracle)

You must specify the following using the TemplateConfiguration API:

- **Data source**—Specify the data source type as `JDBC` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Database type**—You must specify the database type as `oracle`.
- **SQL query**—Specify SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials you created in your Amazon RDS (Oracle) account. The secret is stored in a JSON structure with the following keys:

```
{
    "user name": "database user name",
```

```
    "password": "password"
}
```

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Amazon RDS (Oracle) connector and Amazon Kendra. For more information, see IAM roles for Amazon RDS (Oracle) data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include specific content using user IDs, groups, source URLs, time stamps, and time zones.
- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Amazon RDS (Oracle) data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Amazon RDS (Oracle) template schema.

Notes

- Deleted database rows will not be tracked in when Amazon Kendra checks for updated content.
- The size of field names and values in a row of your database can't exceed 400KB.
- If you have a large amount of data in your database data source, and do not want Amazon Kendra to index all your database content after the first sync, you can choose to sync only new, modified, or deleted documents.
- As a best practice, provide Amazon Kendra with read-only database credentials.
- As a best practice, avoid adding tables with sensitive data or personal identifiable information (PII).

Amazon RDS (PostgreSQL)

Note

Amazon RDS (PostgreSQL) connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

Amazon RDS is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. If you are a Amazon RDS user, you can use Amazon Kendra to index your Amazon RDS (PostgreSQL) data source. The Amazon Kendra Amazon RDS (PostgreSQL) data source connector supports PostgreSQL 9.6.

You can connect Amazon Kendra to your Amazon RDS (PostgreSQL) data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Amazon RDS (PostgreSQL) data source connector, see Troubleshooting data sources.

Topics

Supported features

- Field mappings
- User context filtering
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Amazon RDS (PostgreSQL) data source, make these changes in your Amazon RDS (PostgreSQL) and AWS accounts.

In Amazon RDS (PostgreSQL), make sure you have:

- Noted your database user name and password.

> **Important**
>
> As a best practice, provide Amazon Kendra with read-only database credentials.

- Copied your database host url, port, and instance. You can find this information on the Amazon RDS console.
- Checked each document is unique in Amazon RDS (PostgreSQL) and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Amazon RDS (PostgreSQL) authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Amazon RDS (PostgreSQL) data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Amazon RDS (PostgreSQL) data source you must provide details of your Amazon RDS (PostgreSQL) credentials so that Amazon Kendra can access your data. If you have not yet configured Amazon RDS (PostgreSQL) for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Amazon RDS (PostgreSQL)

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Amazon RDS (PostgreSQL) connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Amazon RDS (PostgreSQL) connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. In **Source**, enter the following information:

   b. **Host** – Enter the database host URL, for example: `http://`*`instance URL`*`.`*`region`*`.rds.amazonaws.com`.

   c. **Port** – Enter the database port, for example, `5432`.

   d. **Instance** – Enter the database instance, for example `postgres`.

   e. **Enable SSL certificate location**—Choose to enter the Amazon S3 path to your SSL certificate file.

   f. In **Authentication**—enter the following information:

      - **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Amazon RDS (PostgreSQL) authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

A. Enter following information in the **Create an AWS Secrets Manager secret window**:

    I. **Secret name**—A name for your secret. The prefix ‘AmazonKendra-Amazon RDS (PostgreSQL)-’ is automatically added to your secret name.

    II. For **Database user name**, and **Password**—Enter the authentication credential values you copied from your database.

B. Choose **Save**.

g. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

    a. In **Sync scope**, choose from the following options :

    - **SQL query**—Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Kendra will crawl all database content that matches your query.
    - **Primary key column**—Provide the primary key for the database table. This identifies a table within your database.
    - **Title column**—Provide the name of the document title column within your database table.
    - **Body column**—Provide the name of the document body column within your database table.

b. In **Additional configuration – *optional***, choose from the following options to sync specific content instead of syncing all files:

- **Change-detecting columns—**Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns.
- **Users' IDs column—**Enter the name of the column which contains User IDs to be allowed access to content.
- **Groups column—**Enter the name of the column that contains groups to be allowed access to content.
- **Source URLs column—**Enter the name of the column which contains Source URLs to be indexed.
- **Time stamps column—**Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
- **Time zones column—**Enter the name of the column which contains time zones for the content to be crawled.
- **Time stamps format—**Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

c. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency**—How often Amazon Kendra will sync with your data source.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. Select from the generated default data source fields—**Document IDs**, **Document titles**, and **Source URLs**—you want to map to Amazon Kendra index.

   b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Amazon RDS (PostgreSQL)

You must specify the following using the [TemplateConfiguration](#) API:

- **Data source**—Specify the data source type as `JDBC` when you use the [TemplateConfiguration](#) JSON schema. Also specify the data source as `TEMPLATE` when you call the [CreateDataSource](#) API.
- **Database type**—You must specify the database type as `postgresql`.
- **SQL query**—Specify SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.

  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials you created in your Amazon RDS (PostgreSQL) account. The secret is stored in a JSON structure with the following keys:

```
{
    "user name": "database user name",
    "password": "password"
}
```

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Amazon RDS (PostgreSQL) connector and Amazon Kendra. For more information, see [IAM roles for Amazon RDS (PostgreSQL) data sources](#).

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see [Configuring Amazon Kendra to use an Amazon VPC](#).
- **Inclusion and exclusion filters**—You can specify whether to include specific content using user IDs, groups, source URLs, time stamps, and time zones.

- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Amazon RDS (PostgreSQL) data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Amazon RDS (PostgreSQL) template schema.

Notes

- Deleted database rows will not be tracked in when Amazon Kendra checks for updated content.
- The size of field names and values in a row of your database can't exceed 400KB.
- If you have a large amount of data in your database data source, and do not want Amazon Kendra to index all your database content after the first sync, you can choose to sync only new, modified, or deleted documents.
- As a best practice, provide Amazon Kendra with read-only database credentials.
- As a best practice, avoid adding tables with sensitive data or personal identifiable information (PII).

Amazon S3

Amazon S3 is an object storage service that stores data as objects within buckets. You can use Amazon Kendra to index your Amazon S3 bucket repository of documents.

> **Warning**
>
> Amazon Kendra doesn't use a bucket policy that grants permissions to an Amazon Kendra principal to interact with an S3 bucket. Instead, it uses IAM roles. Make sure that Amazon Kendra isn't included as a trusted member in your bucket policy to avoid any data security issues in accidentally granting permissions to arbitrary principals. However, you can add a bucket policy to use an Amazon S3 bucket across different accounts. For more information, see Policies to use Amazon S3 across accounts (within the S3 IAM roles tab, under **IAM roles for data sources**). For information about IAM roles for S3 data sources, see IAM roles.

> **Note**
>
> Amazon Kendra now supports an upgraded Amazon S3 connector.
> The console has been automatically upgraded for you. Any new connectors you create in the console will use the upgraded architecture. If you use the API, you must now use the TemplateConfiguration object instead of the `S3DataSourceConfiguration` object to configure your connector.
> Connectors configured using the older console and API architecture will continue to function as configured. However, you won't be able to edit or update them. If you want to edit or update your connector configuration, you must create a new connector.
> We recommended migrating your connector workflow to the upgraded version. Support for connectors configured using the older architecture is scheduled to end by June 2024.

You can connect to your Amazon S3 data source using the the Amazon Kendra console or the TemplateConfiguration API.

> **Note**
>
> To generate a sync status report for your Amazon S3 data source, see Troubleshooting data sources.

For troubleshooting your Amazon Kendra S3 data source connector, see Troubleshooting data sources.

Topics

Supported features

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your S3 data source, make these changes in your S3 and AWS accounts.

In S3, make sure you have:

- Copied the name of your Amazon S3 bucket.

> **Note**
>
> Your bucket must be in the same region as your Amazon Kendra index and your index must have permission to access the bucket that contains your documents.

- Checked each document is unique in S3 and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

If you don't have an existing IAM role, you can use the console to create a new IAM role when you connect your S3 data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and an index ID.

Connection instructions

To connect Amazon Kendra to your S3 data source, you must provide the necessary details of your S3 data source so that Amazon Kendra can access your data. If you have not yet configured S3 for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to Amazon S3

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **S3 connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **S3 connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following optional information:

a. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

b. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

c. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. For **Data source location**—Specify the path to the Amazon S3 bucket where your data is stored. Select **Browse S3** to choose your S3 bucket.

b. For **Maximum file size**—Specify a limit in MB to only crawl files under this limit. The maximum file size Amazon Kendra can allow is 50 MB.

c. For (Optional) **Metadata files prefix folder location**—Specify the path to the folder in which your fields/attributes and other document metadata is stored. Select **Browse S3** to locate your metadata folder.

d. For (Optional) **Access control list configuration file location**—Specify the path to the file that contains a JSON structure of your users and their access to documents. Select **Browse S3** to locate your ACL file.

e. (Optional) **Select decryption key**—Select to use a decryption key. You can choose to use an existing AWS KMS key.

f. For (Optional) **Additional configuration**—Add patterns to include or exclude certain files. All paths are relative to the data source location S3 bucket.

g. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

   - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
   - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

h. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

i. Choose **Next**.

8. On the **Set field mappings** page, enter the following optional information:

   a. **Default field mappings**—Select from the Amazon Kendra generated default data source fields you want to map to your index.

   b. **Add field**—Choose to add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Amazon S3

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as S3 when you use the TemplateConfiguration JSON schema. Also specify the data source as TEMPLATE when you call the CreateDataSource API.
- **BucketName**—The name of the bucket that contains the documents.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the S3 connector and Amazon Kendra. For more information, see IAM roles for S3 data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain file names, file types, file paths. You use glob patterns (patterns that can expand a wildcard pattern into a list of path names that match the given pattern). For examples, see Use of Exclude and Include Filters in the AWS CLI Command Reference.
- **Document metadata and access control configuration**—Add document metadata and access control files that contain information such as the source URI, document author, or custom document attributes/fields, and your users and which documents they can access. Each metadata file contains metadata about a single document.
- **Field mappings**—Choose to map your S3 data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see S3 template schema.

Learn more

To learn more about integrating Amazon Kendra with your S3 data source, see:

- Search for answers accurately using Amazon Kendra S3 Connector with VPC support

Creating an Amazon S3 data source

The following examples demonstrate creating an Amazon S3 data source. The examples assume that you have already created an index and an IAM role with permission to read the data from the index. For more information about the IAM role, see IAM access roles. For more information about creating an index, see Creating an index.

CLI

```
aws kendra create-data-source \
 --index-id index ID \
 --name example-data-source \
 --type S3 \
 --configuration '{"S3Configuration":{"BucketName":"bucket name"}}'
 --role-arn 'arn:aws:iam::account id:role:/role name
```

Python

The following snippet of Python code creates an Amazon S3 data source. For the complete example, see Getting started (AWS SDK for Python (Boto3)).

```
print("Create an Amazon S3 data source.")
```

```
    # Provide a name for the data source
    name = "getting-started-data-source"
    # Provide an optional description for the data source
    description = "Getting started data source."
    # Provide the IAM role ARN required for data sources
    role_arn = "arn:aws:iam::${accountID}:role/${roleName}"
    # Provide the data soource connection information
    s3_bucket_name = "S3-bucket-name"
    type = "S3"
    # Configure the data source
    configuration = {"S3DataSourceConfiguration":
        {
            "BucketName": s3_bucket_name
        }
    }

    data_source_response = kendra.create_data_source(
        Configuration = configuration,
        Name = name,
        Description = description,
        RoleArn = role_arn,
        Type = type,
        IndexId = index_id
    )
```

It can take some time to create your data source. You can monitor the progress by using the [DescribeDataSource](#) API. When the data source status is `ACTIVE` the data source is ready to use.

The following examples demonstrate getting the status of a data source.

CLI

```
aws kendra describe-data-source \
 --index-id index ID \
 --id data source ID
```

Python

The following snippet of Python code gets information about an S3 data source. For the complete example, see [Getting started (AWS SDK for Python (Boto3))](#).

```
print("Wait for Amazon Kendra to create the data source.")
```

```
    while True:
        data_source_description = kendra.describe_data_source(
            Id = "data-source-id",
            IndexId = "index-id"
        )
        status = data_source_description["Status"]
        print(" Creating data source. Status: "+status)
        time.sleep(60)
        if status != "CREATING":
            break
```

This data source doesn't have a schedule, so it doesn't run automatically. To index the data source, you call StartDataSourceSyncJob to synchronize the index with the data source.

The following examples demonstrate synchronizing a data source.

CLI

```
aws kendra start-data-source-sync-job \
 --index-id index ID \
 --id data source ID
```

Python

The following snippet of Python code synchronizes an Amazon S3 data source. For the complete example, see Getting started (AWS SDK for Python (Boto3)).

```
print("Synchronize the data source.")

    sync_response = kendra.start_data_source_sync_job(
        Id = "data-source-id",
        IndexId = "index-id"
    )
```

Amazon S3 document metadata

You can add metadata, additional information about a document, to documents in an Amazon S3 bucket using a metadata file. Each metadata file is associated with an indexed document.

Your metadata files must be stored in the same bucket as your indexed files. You can specify a location within the bucket for your metadata files using the console or the `S3Prefix` field of the `DocumentsMetadataConfiguration` parameter when you create an Amazon S3 data source. If you don't specify an Amazon S3 prefix, your metadata files must be stored in the same location as your indexed documents.

If you specify an Amazon S3 prefix for your metadata files, they are in a directory structure parallel to your indexed documents. Amazon Kendra looks only in the specified directory for your metadata. If the metadata isn't read, check that the directory location matches the location of your metadata.

The following examples show how the indexed document location maps to the metadata file location. Note that the document's Amazon S3 key is appended to the metadata's Amazon S3 prefix and then suffixed with `.metadata.json` to form the metadata file's Amazon S3 path. The combined Amazon S3 key, with the metadata's Amazon S3 prefix and `.metadata.json` suffix must be no more than a total of 1024 characters. It is recommended that you keep your Amazon S3 key below 1000 characters to account for addtional characters when combining your key with the prefix and suffix.

```
Bucket name:
     s3://bucketName
Document path:
     documents
Metadata path:
     none
File mapping
     s3://bucketName/documents/file.txt ->
        s3://bucketName/documents/file.txt.metadata.json
```

```
Bucket name:
     s3://bucketName
Document path:
     documents/legal
Metadata path:
     metadata
File mapping
     s3://bucketName/documents/legal/file.txt ->
        s3://bucketName/metadata/documents/legal/file.txt.metadata.json
```

Your document metadata is defined in a JSON file. The file must be a UTF-8 text file without a BOM marker. The file name of the JSON file must be `<document>.<extension>.metadata.json`. In this example, "document" is the name of the document that the metadata applies to and "extension" is the file extension for the document. The document ID must be unique in `<document>.<extension>.metadata.json`.

The content of the JSON file follows this template. All of the attributes/fields are optional, so it's not necessary to include all attributes. You must provide a value for each attribute you want to include; the value cannot be empty. If you don't specify the `_source_uri`, then the links returned by Amazon Kendra in the search results point to the Amazon S3 bucket that contains the document. `DocumentId` is mapped to the field `s3_document_id` and is the absolute path to the document in S3.

```
{
    "DocumentId": "S3 document ID, the S3 path to doc",
    "Attributes": {
        "_category": "document category",
        "_created_at": "ISO 8601 encoded string",
        "_last_updated_at": "ISO 8601 encoded string",
        "_source_uri": "document URI",
        "_version": "file version",
        "_view_count": number of times document has been viewed,
        "custom attribute key": "custom attribute value",
        additional custom attributes
    },
    "AccessControlList": [
         {
             "Name": "user name",
             "Type": "GROUP | USER",
             "Access": "ALLOW | DENY"
         }
    ],
    "Title": "document title",
    "ContentType": "For example HTML | PDF. For supported content types, see Types of
 documents."
}
```

The `_created_at` and `_last_updated_at` metadata fields are ISO 8601 encoded dates. For example, 2012-03-25T12:30:10+01:00 is the ISO 8601 date-time format for March 25, 2012, at 12:30PM (plus 10 seconds) in the Central European Time time zone.

You can add additional information to the `Attributes` field about a document that you use to filter queries or to group query responses. For more information, see [Creating custom document fields].

You can use the `AccessControlList` field to filter the response from a query. This way, only certain users and groups have access to documents. For more information, see [Filtering on user context].

Access control for Amazon S3 data sources

You can control access to documents in an Amazon S3 data source using a configuration file. You specify the file in the console or as the `AccessControlListConfiguration` parameter when you call the [CreateDataSource] or [UpdateDataSource] API.

The configuration file contains a JSON structure that identifies an S3 prefix and lists the access settings for the prefix. The prefix can be a path, or it can be an individual file. If the prefix is a path, the access settings apply to all of the files in that path. There is a maximum number of S3 prefixes in the JSON configuration file and a default maximum file size. For more information, see [Quotas for Amazon Kendra]

You can specify both users and groups in the access settings. When you query the index, you specify user and group information. For more information, see [Filtering by user attribute].

The JSON structure for the configuration file must be in the following format:

```
[
    {
        "keyPrefix": "s3://BUCKETNAME/prefix1/",
        "aclEntries": [
            {
                "Name": "user1",
                "Type": "USER",
                "Access": "ALLOW"
            },
            {
                "Name": "group1",
                "Type": "GROUP",
                "Access": "DENY"
            }
        ]
    },
```

```
    {
        "keyPrefix": "s3://prefix2",
        "aclEntries": [
            {
                "Name": "user2",
                "Type": "USER",
                "Access": "ALLOW"
            },
            {
                "Name": "user1",
                "Type": "USER",
                "Access": "DENY"
            },
            {
                "Name": "group1",
                "Type": "GROUP",
                "Access": "DENY"
            }
        ]
    }
]
```

Using Amazon VPC with an Amazon S3 data source

This topic provides a step-by-step example that shows how to connect to an Amazon S3 bucket by using an Amazon S3 connector through Amazon VPC. The example assumes that you're starting with an existing S3 bucket. We recommend that you upload just a few documents to your S3 bucket to test the example.

You can connect Amazon Kendra to your Amazon S3 bucket through Amazon VPC. To do so, you must specify the Amazon VPC subnet and Amazon VPC security groups when creating your Amazon S3 data source connector.

> **Important**
>
> So that an Amazon Kendra Amazon S3 connector can access your Amazon S3 bucket, make sure that you have assigned an Amazon S3 endpoint to your virtual private cloud (VPC).

For Amazon Kendra to sync documents from your Amazon S3 bucket through Amazon VPC, you must complete the following steps:

- Set up an Amazon S3 endpoint for Amazon VPC. For more information about how to set up an Amazon S3 endpoint, see Gateway endpoints for Amazon S3 in the *AWS PrivateLink Guide*.
- (Optional) Checked your Amazon S3 bucket policies to make sure that the Amazon S3 bucket is accessible from the virtual private cloud (VPC) that you assigned to Amazon Kendra. For more information, see Controlling access from VPC endpoints with bucket policies in the *Amazon S3 User Guide*

Steps

Step 1: Configure an Amazon VPC

Create a VPC network including a private subnet with an Amazon S3 gateway endpoint and a security group for Amazon Kendra to use later.

To configure a VPC with a private subnet, an S3 endpoint, and a security group

1. Sign in to the AWS Management Console and open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. **Create a VPC with a private subnet and an S3 endpoint for Amazon Kendra to use:**

   From the navigation pane, choose **Your VPCs**, and then choose **Create VPC**.

   a. For **Resources to create**, choose **VPC and more**.
   b. For **Name tag**, enable **Auto-generate**, then enter **`kendra-s3-example`**.
   c. For **IPv4 / IPv6 CIDR block**, keep the default values.
   d. For **Number of Availability Zones (AZs)**, choose **number 1**.
   e. Select **Customize AZs**, and then select an Availability Zone from the **First availability zone** list.

      Amazon Kendra only supports a specific set of Availability Zones.
   f. For **Number of public subnets**, choose **number 0**.
   g. For **Number of private subnets**, choose **number 1**.
   h. For **NAT gateways**, choose **None**.

i. For **VPC endpoints**, choose **Amazon S3 gateway.**.
j. Leave the rest of the values at their default settings.
k. Select **Create VPC**.

   Wait until the **Create VPC** workflow finishes. Then, choose **View VPC** to check the **VPC** you just created.

You have now created a VPC network with a private subnet, which does not have access to the public internet.

3. **Copy your VPC endpoint ID of your Amazon S3 endpoint:**

   a. From the navigation pane, choose **Endpoints**.
   b. In the **Endpoints** list, find the Amazon S3 endpoint `kendra-s3-example-vpce-s3` that you just created together with your VPC.
   c. Make a note of the **VPC endpoint ID**.

   You have now created an Amazon S3 gateway endpoint to access your Amazon S3 bucket through a subnet.

4. **Create a Security Group for Amazon Kendra to use:**

   a. From the navigation pane, choose **Security Groups**, then select **Create security group**.
   b. For **Security group name**, enter **`s3-data-source-security-group`**.
   c. Choose your VPC from the **Amazon VPC** list.
   d. Leave **inbound rules** and **outbound rules** as the default.
   e. Choose **Create security group**.

   You have now created a VPC security group.

You assign the subnet and security group that you created to your Amazon Kendra Amazon S3 data source connector during the connector configuration process.

(Optional) Step 2: Configure Amazon S3 bucket policy

In this optional step, learn how to configure an Amazon S3 bucket policy so that your Amazon S3 bucket is only accessible from the VPC that you assign to Amazon Kendra.

Amazon Kendra uses IAM roles to access your Amazon S3 bucket and doesn't require that you configure an Amazon S3 bucket policy. However, you might find it useful to create a bucket policy if you want to configure an Amazon S3 connector using an Amazon S3 bucket that has existing policies restricting access to it from the public internet.

To configure your Amazon S3 bucket policy

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. From the navigation pane, choose **Buckets**.
3. Choose the name of the Amazon S3 bucket that you want to sync with Amazon Kendra.
4. Choose the **Permissions** tab, scroll down to **Bucket policy**, and then click on **Edit**.
5. Add or modify your bucket policy to allow access only from the VPC endpoint that you created.

   The following is an example bucket policy. Replace *`bucket-name`* and *`vpce-id`* with your Amazon S3 bucket name and the Amazon S3 endpoint ID that you noted earlier.
6. Select **Save changes**.

Your S3 bucket is now accessible only from the specific VPC that you created.

Step 3: Create a test Amazon S3 data source connector

To test your Amazon VPC configuration, create an Amazon S3 connector. Then, configure it with the VPC that you created by following the steps outlined in [Amazon S3](#).

For Amazon VPC configuration values, choose the values that you created during this example:

- **Amazon VPC(VPC)** – `kendra-s3-example-vpc`
- **Subnets** – `kendra-s3-example-subnet-private1-[availability zone]`
- **Security groups** – `s3-data-source-security-group`

Wait for your connector to finish creating. After the Amazon S3 connector has been created, choose **Sync now** to initiate a sync.

It might take several minutes to several hours to finish the sync, depending on how many documents are in your Amazon S3 bucket. To test the example, we recommend that you upload just a few documents to your S3 bucket. If your configuration is correct, you should eventually see a **Sync status** of **Completed**.

If you encounter any errors, see Troubleshooting Amazon VPC connection.

Amazon Kendra Web Crawler

You can use Amazon Kendra Web Crawler to crawl and index web pages.

You can only crawl public facing websites or internal company websites that use the secure communication protocol Hypertext Transfer Protocol Secure (HTTPS). If you receive an error when crawling a website, it could be that the website is blocked from crawling. To crawl internal websites, you can set up a web proxy. The web proxy must be public facing. You can also use authentication to access and crawl websites.

When selecting websites to index, you must adhere to the Amazon Acceptable Use Policy and all other Amazon terms. Remember that you must only use Amazon Kendra Web Crawler to index your own web pages, or web pages that you have authorization to index. To learn how to stop Amazon Kendra Web Crawler from indexing your website(s), please see Configuring the `robots.txt` file for Amazon Kendra Web Crawler.

> **Note**
>
> Abusing Amazon Kendra Web Crawler to aggressively crawl websites or web pages you don't own is **not** considered acceptable use.

Amazon Kendra has two versions of the web crawler connector. Supported features of each version include:

Amazon Kendra Web Crawler connector v1.0 / WebCrawlerConfiguration API

- Web proxy
- Inclusion/exclusion filters

Amazon Kendra Web Crawler connector v2.0 / TemplateConfiguration API

- Field mappings
- Inclusion/exclusion filters
- Full and incremental content syncs
- Web proxy

- Basic, NTLM/Kerberos, SAML, and form authentication for your websites
- Virtual private cloud (VPC)

> **Important**
>
> Web Crawler v2.0 connector creation is not supported by AWS CloudFormation. Use the Web Crawler v1.0 connector if you need AWS CloudFormation support.

For troubleshooting your Amazon Kendra web crawler data source connector, see Troubleshooting data sources.

Topics

- Amazon Kendra Web Crawler connector v1.0
- Amazon Kendra Web Crawler connector v2.0
- Configuring the robots.txt file for Amazon Kendra Web Crawler

Amazon Kendra Web Crawler connector v1.0

You can use Amazon Kendra Web Crawler to crawl and index web pages.

You can only crawl public facing websites and websites that use the secure communication protocol Hypertext Transfer Protocol Secure (HTTPS). If you receive an error when crawling a website, it could be that the website is blocked from crawling. To crawl internal websites, you can set up a web proxy. The web proxy must be public facing.

When selecting websites to index, you must adhere to the Amazon Acceptable Use Policy and all other Amazon terms. Remember that you must only use Amazon Kendra Web Crawler to index your own web pages, or web pages that you have authorization to index. To learn how to stop Amazon Kendra Web Crawler from indexing your website(s), please see Configuring the `robots.txt` file for Amazon Kendra Web Crawler.

> **Note**
>
> Abusing Amazon Kendra Web Crawler to aggressively crawl websites or web pages you don't own is **not** considered acceptable use.

For troubleshooting your Amazon Kendra web crawler data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more

Supported features

- Web proxy
- Inclusion/exclusion filters

Prerequisites

Before you can use Amazon Kendra to index your websites, check the details of your websites and AWS accounts.

For your websites, make sure you have:

- Copied the seed or sitemap URLs of the websites you want to index.
- **For websites that require basic authentication**: Noted the user name and password, and copied the host name of the website and the port number.
- **Optional:** Copied the host name of the website and the port number if you want to use a web proxy to connect to internal websites you want to crawl. The web proxy must be public facing. Amazon Kendra supports connecting to web proxy servers that are backed by basic authentication or you can connect with no authentication.
- Checked each web page document you want to index is unique and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.

- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- For websites that require authentication, or if using a web proxy with authentication, stored your authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your web crawler data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your web crawler data source, you must provide the necessary details of your web crawler data source so that Amazon Kendra can access your data. If you have not yet configured web crawler for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to web crawler

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **web crawler connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **web crawler connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. For **Source**, choose between **Source URLs** and **Source sitemaps** depending on your use case and enter the values for each.

      You can add up to 10 source URLs and three sitemaps.

      > **Note**
      >
      > If you want to crawl a sitemap, check that the base or root URL is the same as the URLs listed on your sitemap page. For example, if your sitemap URL is *https://example.com/sitemap-page.html*, the URLs listed on this sitemap page should also use the base URL "https://example.com/".

   b. (Optional) For **Web proxy**— enter the following information:

i. **Host name**—The host name where web proxy is required.

ii. **Port number**—The port used by the host URL transport protocol. The port number should be a numeric value between 0 and 65535.

iii. For **Web proxy credentials**—If your web proxy connection requires authentication, choose an existing secret or create a new secret to store your authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

iv. Enter the following information in the **Create an AWS Secrets Manager Secrets Manager secret window**:

A. **Secret name**—A name for your secret. The prefix 'AmazonKendra-WebCrawler-' is automatically added to your secret name.

B. For **User name** and **Password**—Enter these basic authentication credentials for your websites.

C. Choose **Save**.

c. (Optional) **Hosts with authentication**—Select to add additional hosts with authentication.

d. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

e. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. **Crawl range**—Choose the kind of web pages you want to crawl.

b. **Crawl depth**—Select number of levels from the seed URL that Amazon Kendra should crawl.

c. **Advanced crawl settings** and **Additional configuration**enter the following information:

i. **Maximum file size**—The maximum web page or attachment size to crawl. Minimum 0.000001 MB (1 byte). Maximum 50 MB.

ii. **Maximum links per page**—The maximum number of links crawled per page. Links are crawled in order of appearance. Minimum 1 link/page. Maximum 1000 links/page.

iii. **Maximum throttling**—The maximum number of URLs crawled per host name per minute. Minimum 1 URLs/host name/minute. Maximum 300 URLs/host name/minute.

iv. **Regex patterns**—Add regular expression patterns to include or exclude certain URLs. You can add up to 100 patterns.

d. In **Sync run schedule**, for **Frequency**—Choose how often Amazon Kendra will sync with your data source.

e. Choose **Next**.

8. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to web crawler

You must specify the following using the WebCrawlerConfiguration API:

- **URLs**—Specify the seed or starting point URLs of the websites or the sitemap URLs of the websites you want to crawl using SeedUrlConfiguration and SiteMapsConfiguration.

> **Note**
>
> If you want to crawl a sitemap, check that the base or root URL is the same as the URLs listed on your sitemap page. For example, if your sitemap URL is *https://example.com/sitemap-page.html*, the URLs listed on this sitemap page should also use the base URL "https://example.com/".

- **Secret Amazon Resource Name (ARN)**—If a website requires basic authentication, you provide the host name, port number and a secret that stores your basic authentication credentials of your user name and password. You provide the secret ARN using the AuthenticationConfiguration API. The secret is stored in a JSON structure with the following keys:

```
{
    "username": "user name",
    "password": "password"
}
```

You can also provide web proxy credentials using an AWS Secrets Manager secret. You use the ProxyConfiguration API to provide the website host name and port number, and optionally the secret that stores your web proxy credentials.

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the web crawler connector and Amazon Kendra. For more information, see IAM roles for web crawler data sources.

You can also add the following optional features:

- **Crawl mode**—Choose whether to crawl website host names only, or host names with subdomains, or also crawl other domains the web pages link to.
- The 'depth' or number of levels from the seed level to crawl. For example, the seed URL page is depth 1 and any hyperlinks on this page that are also crawled are depth 2.
- The maximum number of URLs on a single web page to crawl.
- The maximum size in MB of a web page to crawl.
- The maximum number of URLs crawled per website host per minute.
- The web proxy host and port number to connect to and crawl internal websites. For example, the host name of *https://a.example.com/page1.html* is "a.example.com" and the port number is is 443, the standard port for HTTPS. If web proxy credentials are required to connect to a website host, you can create an AWS Secrets Manager that stores the credentials.
- The authentication information to access and crawl websites that require user authentication.
- You can extract HTML meta tags as fields using the *Custom Document Enrichment* tool. For more information, see Customizing document metadata during the ingestion process. For an example of extracting HTML meta tags, see CDE examples.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain URLs.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn’t match the inclusion filter isn’t indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

Learn more

To learn more about integrating Amazon Kendra with your web crawler data source, see:

- Reimagine knowledge discovery using Amazon Kendra's Web Crawler

Amazon Kendra Web Crawler connector v2.0

You can use Amazon Kendra Web Crawler to crawl and index web pages.

You can only crawl public facing websites or internal company websites that use the secure communication protocol Hypertext Transfer Protocol Secure (HTTPS). If you receive an error when crawling a website, it could be that the website is blocked from crawling. To crawl internal websites, you can set up a web proxy. The web proxy must be public facing. You can also use authentication to access and crawl websites.

Amazon Kendra Web Crawler v2.0 uses the Selenium web crawler package and a Chromium driver. Amazon Kendra automatically updates the version of Selenium and the Chromium driver using Continuous Integration (CI).

When selecting websites to index, you must adhere to the Amazon Acceptable Use Policy and all other Amazon terms. Remember that you must only use Amazon Kendra Web Crawler to index your own web pages, or web pages that you have authorization to index. To learn how to stop Amazon Kendra Web Crawler from indexing your website(s), please see Configuring the `robots.txt` file for Amazon Kendra Web Crawler.. Abusing Amazon Kendra Web Crawler to aggressively crawl websites or web pages you don't own is **not** considered acceptable use.

For troubleshooting your Amazon Kendra web crawler data source connector, see Troubleshooting data sources.

> **Note**
>
> Web Crawler connector v2.0 does *not* support crawling web site lists from AWS KMS encrypted Amazon S3 buckets. It supports only server-side encryption with Amazon S3 managed keys.

> **Important**
>
> Web Crawler v2.0 connector creation is not supported by AWS CloudFormation. Use the Web Crawler v1.0 connector if you need AWS CloudFormation support.

Topics

Supported features

- Field mappings
- Inclusion/exclusion filters
- Full and incremental content syncs
- Web proxy
- Basic, NTLM/Kerberos, SAML, and form authentication for your websites
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your websites, check the details of your websites and AWS accounts.

For your websites, make sure you have:

- Copied the seed or sitemap URLs of the websites you want to index. You can store the URLs in a text file and upload this to an Amazon S3 bucket. Each URL in the text file must be formatted on a separate line. If you want to store your sitemaps in an Amazon S3 bucket, make sure you have copied the sitemap XML and saved this in an XML file. You can also club multiple sitemap XML files into a ZIP file.

> **Note**
>
> (On-premise/server) Amazon Kendra checks if the endpoint information included in AWS Secrets Manager is the same the endpoint information specified in your data source configuration details. This helps protect against the confused deputy problem, which is a security issue where a user doesn't have permission to perform an action but uses Amazon Kendra as a proxy to access the configured secret and perform the action. If you later change your endpoint information, you must create a new secret to sync this information.

- **For websites that require basic, NTLM, or Kerberos authentication**:
  - Noted your website authentication credentials, which include a user name and password.

  > **Note**
  >
  > Amazon Kendra Web Crawler v2.0 supports the NTLM authentication protocol that includes password hashing, and Kerberos authentication protocol that includes password encryption.

- **For websites that require SAML or login form authentication**:
  - Noted your website authentication credentials, which include a user name and password.
  - Copied the XPaths (XML Path Language) of the user name field (and the user name button if using SAML), password field and button, and copied the login page URL. You can find the XPaths of elements using your web browser's developer tools. XPaths usually follow this format: `//tagname[@Attribute='Value']`.

  > **Note**
  >
  > Amazon Kendra Web Crawler v2.0 uses a headless Chrome browser and the information from the form to authenticate and authorize access with an OAuth 2.0 protected URL.

- **Optional**: Copied the host name and the port number of the web proxy server if you want to use a web proxy to connect to internal websites you want to crawl. The web proxy must be public facing. Amazon Kendra supports connecting to web proxy servers that are backed by basic authentication or you can connect with no authentication.
- **Optional**: Copied the virtual private cloud (VPC) subnet ID if you want to use a VPC to connect to internal websites you want to crawl. For more information, see Configuring an Amazon VPC.
- Checked each web page document you want to index is unique and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the Amazon Resource Name of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- For websites that require authentication, or if using a web proxy with authentication, stored your authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your web crawler data source to Amazon Kendra.

If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your web crawler data source, you must provide the necessary details of your web crawler data source so that Amazon Kendra can access your data. If you have not yet configured web crawler for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to web crawler

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **web crawler connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **web crawler connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:
    a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.
    b. (Optional) **Description**—Enter an optional description for your data source.
    c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.
    d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.
    e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

a. **Source**—Choose either **Source URLs**, **Source sitemaps**, **Source URLs file**, **Source sitemaps file**. If you choose to use a text file that includes a list of up to 100 seed URLs, you specify the path to the Amazon S3 bucket where your file is stored. If you choose to use a sitemap XML file, you specify the path to the Amazon S3 bucket where your file is stored. You can also club multiple sitemap XML files into a ZIP file. Otherwise, you can manually enter up to 10 seed or starting point URLs, and up to three sitemap URLs.

> **Note**
>
> If you want to crawl a sitemap, check that the base or root URL is the same as the URLs listed on your sitemap page. For example, if your sitemap URL is *https://example.com/sitemap-page.html*, the URLs listed on this sitemap page should also use the base URL "https://example.com/".

If your websites require authentication to access the websites, you can choose ether basic, NTLM/Kerberos, SAML, or form authentication. Otherwise, choose the option for no authentication.

> **Note**
>
> If you want to later edit your data source to change your seed URLs with authentication to sitemaps, you must create a new data source. Amazon Kendra configures the data source using the seed URLs endpoint information in the Secrets Manager secret for authentication, and therefore cannot re-configure the data source when changing to sitemaps.

- **AWS Secrets Manager secret**—If your websites require the same authentication to access the websites, choose an existing secret or create a new Secrets Manager secret to store your website credentials. If you choose to create a new secret, an AWS Secrets Manager secret window opens.

  If you chose **Basic** or **NTML/Kerberos** authentication, enter a name for the secret, plus the user name and password. NTLM authentication protocol includes password hashing, and Kerberos authentication protocol includes password encryption.

If you chose **SAML** or **Form** authentication, enter a name for the secret, plus the user name and password. Use XPath for the user name field (and XPath for the user name button if using SAML). Use XPaths for the password field and button, and login page URL. You can find the XPaths (XML Path Language) of elements using your web browser's developer tools. XPaths usually follow this format: `//tagname[@Attribute='Value']`.

b. (Optional) **Web proxy**—Enter the host name and the port number of the proxy sever you want to use to connect to internal websites. For example, the host name of *https://a.example.com/page1.html* is "a.example.com" and the port number is is 443, the standard port for HTTPS. If web proxy credentials are required to connect to a website host, you can create an AWS Secrets Manager that stores the credentials.

c. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

d. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

e. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. **Sync scope**—Set limits for crawling web pages including their domains, file sizes and links; and filter URLs using regex patterns.

      i. (Optional) **Crawl domain range**—Choose whether to crawl website domains only, domains with subdomains, or also crawl other domains that the web pages link to. By default, Amazon Kendra only crawls the domains of the websites you want to crawl.

      ii. (Optional) **Additional configuration**—Set the following settings:

  - **Crawl depth—**The 'depth' or number of levels from the seed level to crawl. For example, the seed URL page is depth 1 and any hyperlinks on this page that are also crawled are depth 2.
  - **Maximum file size—**The maximum size in MB of a web page or attachment to crawl.
  - **Maximum links per page—**The maximum number of URLs on a single webpage to crawl.
  - **Maximum throttling of crawling speed—**The maximum number of URLs crawled per website host per minute.
  - **Files—**Choose to crawl files that the web pages link to.
  - **Crawl and index URLs—**Add regular expression patterns to include or exclude crawling certain URLs, and indexing any hyperlinks on these URL web pages.

   b. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

      - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
      - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

   c. **Sync run schedule—**For **Frequency**, choose how often Amazon Kendra will sync with your data source.

   d. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. Select from the Amazon Kendra generated default fields of web pages and files that you want to map to your index.

   b. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this

page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to web crawler

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `WEBCRAWLERV2` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **URLs**—Specify the seed or starting point URLs of the websites or the sitemap URLs of the websites you want to crawl. You can specify the path to an Amazon S3 bucket that stores your list of seed URLs. Each URL in the text file for seed URLs must be formatted on a separate line. You can also specify the path to an Amazon S3 bucket that stores your sitemap XML files. You can club together multiple sitemap files into a ZIP file and store the ZIP file in your Amazon S3 bucket.

> **Note**
>
> If you want to crawl a sitemap, check that the base or root URL is the same as the URLs listed on your sitemap page. For example, if your sitemap URL is *https://example.com/sitemap-page.html*, the URLs listed on this sitemap page should also use the base URL "https://example.com/".

- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

- **Authentication**—If your websites require the same authentication, specify either `BasicAuth`, `NTLM_Kerberos`, `SAML`, or `Form` authentication. If your websites don't require authentication, specify `NoAuthentication`.
- **Secret Amazon Resource Name (ARN)**—If your websites require basic, NTLM, or Kerberos authentication, you provide a secret that stores your authentication credentials of your user name and password. You provide the Amazon Resource Name (ARN) of an AWS Secrets Manager secret. The secret is stored in a JSON structure with the following keys:

```
{
    "seedUrlsHash": "Hash representation of all seed URLs",
    "userName": "user name",
    "password": "password"
}
```

If your websites require SAML authentication, the secret is stored in a JSON structure with the following keys:

```
{
    "seedUrlsHash": "Hash representation of all seed URLs",

    "userName": "user name",
    "password": "password",
    "userNameFieldXpath": "XPath for user name field",
    "userNameButtonXpath": "XPath for user name button",
    "passwordFieldXpath": "XPath for password field",
    "passwordButtonXpath": "XPath for password button",
    "loginPageUrl": "Full URL for website login page"
}
```

If your websites require form authentication, the secret is stored in a JSON structure with the following keys:

```
{
    "seedUrlsHash": "Hash representation of all seed URLs",
    "userName": "user name",
    "password": "password",
    "userNameFieldXpath": "XPath for user name field",
    "passwordFieldXpath": "XPath for password field",
    "passwordButtonXpath": "XPath for password button",
    "loginPageUrl": "Full URL for website login page"
```

```
}
```

You can find the XPaths (XML Path Language) of elements using your web browser's developer tools. XPaths usually follow this format: `//tagname[@Attribute='Value']`.

You can also provide web proxy credentials using and AWS Secrets Manager secret.

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the web crawler connector and Amazon Kendra. For more information, see IAM roles for web crawler data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Domain range**—Choose whether to crawl website domains with subdomains only, or also crawl other domains the web pages link to. By default, Amazon Kendra only crawls the domains of the websites you want to crawl.
- The 'depth' or number of levels from the seed level to crawl. For example, the seed URL page is depth 1 and any hyperlinks on this page that are also crawled are depth 2.
- The maximum number of URLs on a single web page to crawl.
- The maximum size in MB of a web page or attachment to crawl.
- The maximum number of URLs crawled per website host per minute.
- The web proxy host and port number to connect to and crawl internal websites. For example, the host name of *https://a.example.com/page1.html* is "a.example.com" and the port number is is 443, the standard port for HTTPS. If web proxy credentials are required to connect to a website host, you can create an AWS Secrets Manager that stores the credentials.
- **Inclusion and exclusion filters**—Specify whether to include or exclude crawling certain URLs and indexing any hyperlinks on these URL web pages.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn’t match the

inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Field mappings**—Choose to map the fields of web pages and web page files to your Amazon Kendra index fields. For more information, see Mapping data source fields.

For a list of other important JSON keys to configure, see Amazon Kendra Web Crawler template schema.

Configuring the `robots.txt` file for Amazon Kendra Web Crawler

Amazon Kendra is an intelligent search service that AWS customers use to index and search documents of their choice. In order to index documents on the web, customers may use Amazon Kendra Web Crawler, indicating which URL(s) should be indexed and other operational parameters. Amazon Kendra customers are required to obtain authorization before indexing any particular website.

Amazon Kendra Web Crawler respects standard robots.txt directives like `Allow` and `Disallow`. You can modify the `robots.txt` file of your website to control how Amazon Kendra Web Crawler crawls your website.

Configuring how Amazon Kendra Web Crawler accesses your website

You can control how the Amazon Kendra Web Crawler indexes your website using `Allow` and `Disallow` directives. You can also control which web pages are indexed and which web pages are not crawled.

To allow Amazon Kendra Web Crawler to crawl all web pages except disallowed web pages, use the following directive:

```
User-agent: amazon-kendra    # Amazon Kendra Web Crawler
Disallow: /credential-pages/ # disallow access to specific pages
```

To allow Amazon Kendra Web Crawler to crawl only specific web pages, use the following directive:

```
User-agent: amazon-kendra    # Amazon Kendra Web Crawler
Allow: /pages/ # allow access to specific pages
```

To allow Amazon Kendra Web Crawler to crawl all website content and disallow crawling for any other robots, use the following directive:

```
User-agent: amazon-kendra # Amazon Kendra Web Crawler
Allow: / # allow access to all pages
User-agent: * # any (other) robot
Disallow: / # disallow access to any pages
```

Stopping Amazon Kendra Web Crawler from crawling your website

You can stop Amazon Kendra Web Crawler from indexing your website using the `Disallow` directive. You can also control which web pages are crawled and which are not.

To stop Amazon Kendra Web Crawler from crawling the website, use the following directive:

```
User-agent: amazon-kendra # Amazon Kendra Web Crawler
Disallow: / # disallow access to any pages
```

If you have any questions or concerns regarding Amazon Kendra Web Crawler, you can reach out to the AWS support team.

Box

Box is a cloud storage service that offers file hosting capabilities. You can use Amazon Kendra to index content in your Box content, including comments, tasks, and web links.

You can connect Amazon Kendra to your Box data source using the Amazon Kendra console and the BoxConfiguration API.

For troubleshooting your Amazon Kendra Box data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more
- Notes

Supported features

Amazon Kendra Box data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters
- Change log, full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Box data source, make these changes in your Box and AWS accounts.

In Box, make sure you have:

- A Box Enterprise or Box Enterprise Plus account.
- Configured a Box custom app in the Box Developer Console, with Server-side authentication using JSON Web Tokens (JWT). See Box documentation on creating a Custom App and Box documentation of configuring JWT Auth for more details.
- Set your **App Access Level** to **App + Enterprise Access** and allowed it to **Make API calls using the as-user header**.
- Used the admin user to add the following **Application Scopes** in your Box app:
  - Write all files and folders stored in a Box
  - Manage users
  - Manage groups
  - Manage enterprise properties
- Configured Public/Private key pair including a client ID, a client secret, a public key ID, private key ID, a pass phrase, and an enterprise ID to use as your authentication credentials. See Public and private key pair for more details.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you

re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Copied your Box enterprise ID either from your Box Developer Console settings or from your Box app. For example, *801234567*.
- Checked each document is unique in Box and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Box authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Box data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Box data source, you must provide the necessary details of your Box data source so that Amazon Kendra can access your data. If you have not yet configured Box for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to Box

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Box connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Box connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. **Box enterprise ID**—Enter your Box Enterprise ID. For example, *801234567*.

b. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

c. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Box authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

   i. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Box-' is automatically added to your secret name.

   ii. For **Client ID**, **Client Secret**, **Public Key ID**, **Private Key ID**, and **Pass Phrase**—Enter the values from the Public/Private Key you configured in Box.

   iii. Add and save your secret.

d. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

e. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

f. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

g. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. **Box files—**Choose whether to crawl web links, comments, and tasks.

b. For **Additional configuration—**Add regular expression patterns to include or exclude certain content.

c. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

   - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
   - New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
   - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule** for **Frequency—**Choose how often to sync your data source content and update your index.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. **Default data source fields—**Select from the Amazon Kendra generated default data source fields you want to map to your index.

   b. **Add field—**To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Box

You must specify the following using the BoxConfiguration API:

Box enterprise ID—Provide your Box Enterprise ID. You can find the enterprise ID in the Box Developer Console settings or when you configure an app in Box.

- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Box account. The secret is stored in a JSON structure with the following keys:

```
{
    "clientID": "client-id",
    "clientSecret": "client-secret",
    "publicKeyID": "public-key-id",
    "privateKey": "private-key",
    "passphrase": "pass-phrase"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Box connector and Amazon Kendra. For more information, see IAM roles for Box data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` as part of the data source configuration. See Configuring Amazon Kendra to use a VPC.
- **Change log**—Whether Amazon Kendra should use the Box data source change log mechanism to determine if a document must be updated in the index.

> **Note**
>
> Use the change log if you don't want Amazon Kendra to scan all of the documents. If your change log is large, it might take Amazon Kendra less time to scan the documents in the Box data source than to process the change log. If you are syncing your Box data source with your index for the first time, all documents are scanned.

- **Comments, tasks, web links**—Specify whether to crawl these types of content.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Inclusion and exclusion filters**—Specify whether to include or exclude certain Box files and folders.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Box data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

Learn more

To learn more about integrating Amazon Kendra with your Box data source, see:

- [Getting started with the Amazon Kendra Box connector]

Notes

- When Access Control Lists (ACLs) are enabled, the "Sync only new or modified content" option is not available due to Box API limitations. We recommend using "Full sync" or "New, modified, or deleted content sync" modes instead, or disable ACLs if you need to use this sync mode.

Confluence

Confluence is a collaborative work-management tool designed for sharing, storing, and working on project planning, software development, and product management. Amazon Kendra supports both Confluence Server/Data Center and Confluence Cloud. You can use Amazon Kendra to index the following Confluence entities:

- **Spaces –** Top-level designated areas for organizing related content. Each space serves as a container, capable of holding multiple pages, blogs, and attachments.
- **Pages –** Individual documents within a space where users create and manage content. Pages can contain text, images, tables, and multimedia elements, and can have nested sub-pages. Each page is considered a single document.
- **Blogs –** Content similar to pages, typically used for updates or announcements. Each blog post is considered as a single document.
- **Comments** – Allows users to give feedback or engage in discussions on specific content within pages or blog posts.
- **Attachments –** Files uploaded to pages or blog posts in Confluence, such as images, documents, or other file types.

By default, Amazon Kendra doesn't index Confluence archives and personal spaces. You can choose to index them when you create the data source. If you don't want Amazon Kendra to index a space, mark it private in Confluence.

You can connect Amazon Kendra to your Confluence data source using either the Amazon Kendra console, the TemplateConfiguration API, or the ConfluenceConfiguration API.

Amazon Kendra has two versions of the Confluence connector. The following features are supported.

Confluence connector V2.0 / TemplateConfiguration API

- Field mappings
- User access control
- Inclusion/exclusion patterns
- Full and incremental content syncs
- Virtual private cloud (VPC)

Confluence connector V1.0 / ConfluenceConfiguration API (no longer supported)

- Field mappings
- User access control
- Inclusion/exclusion filters
- (Confluence Server only) Virtual private cloud (VPC)

> **Note**
>
> Confluence connector V1.0 / ConfluenceConfiguration API ended in 2023. We recommend migrating to or using Confluence connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra Confluence data source connector, see Troubleshooting data sources.

Topics

ACLs in Confluence Connector

Connectors support crawling Access Control Lists (ACLs) and identifying information where applicable based on the data source. If you index documents without ACLs, all documents are considered public. Indexing documents with ACLs ensures data security.

The Amazon Kendra Confluence connector scans spaces to collect pages and blog posts along with their ACLs. If there is no restriction applied on a page or blog, the connector inherits permissions from its space. If specific user or group restriction is applied on a page, only those users will be able to access that page. If page is nested, the nested page inherits the permissions of parent page if no restrictions are applied. A similar permissions model applies to blogs; however, Confluence does not support nested blogs.

In addition, Amazon Kendra Confluence connector crawls user principal information (local user alias, local group and federated group identity configurations) and its permissions for each configured space.

> **Note**
>
> The Confluence Cloud connector does not support crawling macros, whiteboards, or databases.

The Amazon Kendra Confluence connector updates ACL changes each time it crawls your data source content. To ensure the correct users have access to the correct content, regularly re-sync your data source to capture any ACL updates.

Confluence connector V2.0

Confluence is a collaborative work-management tool designed for sharing, storing, and working on project planning, software development, and product management. You can use Amazon Kendra to index your Confluence spaces, pages (including nested pages), blogs, and comments and attachments to indexed pages and blogs.

For troubleshooting your Amazon Kendra Confluence data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions

Supported features

Amazon Kendra Confluence data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion patterns
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Confluence data source, make these changes in your Confluence and AWS accounts.

In Confluence, make sure you have:

- Copied your Confluence instance URL. For example: *https://example.confluence.com*, or *https://www.example.confluence.com/*, or *https:// atlassian.net/*. You need your Confluence instance URL to connect to Amazon Kendra.

  If you're using Confluence Cloud, your host URL must end with *atlassian.net/*.

> **Note**
>
> The following URL formats are **not** supported:
>
> - *https://example.confluence.com/xyz*
> - *https://www.example.confluence.com//wiki/spacekey/xxx*
> - *https://atlassian.net/xyz*

> **Note**
>
> (On-premise/server) Amazon Kendra checks if the endpoint information included in AWS Secrets Manager is the same the endpoint information specified in your data source configuration details. This helps protect against the confused deputy problem, which is a security issue where a user doesn't have permission to perform an action but uses Amazon Kendra as a proxy to access the configured secret and perform the action. If you later change your endpoint information, you must create a new secret to sync this information.

- Configured basic authentication credentials containing a user name (email ID used to log into Confluence) and password (Confluence API token as the password). See [Manage API tokens for your Atlassian account]().

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **Optional:** Configured OAuth 2.0 credentials containing a Confluence app key, Confluence app secret, Confluence access token, and Confluence refresh token to allow Amazon Kendra to connect to your Confluence instance. If your access token expires, you can either use the refresh token to regenerate your access token and refresh token pair. Or, you can repeat the authorization process. For more information on access tokens, see Manage OAuth access tokens.
- (For Confluence Server/Data Center only) **Optional:** Configured a Personal Access Token (PAT) in Confluence. See Using Personal Access Tokens.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Confluence authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you

re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Confluence data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Confluence data source, you must provide the necessary details of your Confluence data source so that Amazon Kendra can access your data. If you have not yet configured Confluence for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Confluence

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Confluence connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Confluence connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. In **Source**, choose either **Confluence Cloud** or **Confluence Server/Data Center**.

   b. **Confluence URL**—Enter the Confluence host URL. For example, *`https://example.confluence.com`*.

   c. (For Confluence Server/Data Center only) **SSL certificate location - *optional***—Enter the Amazon S3 path to your SSL certificate file for Confluence Server.

   d. (For Confluence Server/Data Center only) **Web proxy - *optional***—Enter the web proxy host name (without the `http://` or `https://` protocol) and port number (port used by the host URL transport protocol). The port number should be a numeric value between 0 and 65535.

   e. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

   f. **Authentication**—Choose either **Basic authentication**, **Oauth 2.0 authentication**, or (For Confluence Server/Data Center only) **Personal Access Token authentication**.

   g. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Confluence authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens. Enter the following information in the window:

      i. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Confluence-' is automatically added to your secret name.

      ii. If using **Basic Authentication**—Enter the secret name, user name, and password (Confluence API token as the password) you configured in Confluence.

If using **OAuth2.0 Authentication**—Enter the secret name, app key, app secret, access token, and refresh token you configured in Confluence.

(Confluence Server/Data Center only) If using **Personal Access Token authentication**—Enter the secret name and Confluence token you configured in your Confluence.

iii. Save and add your secret.

h. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

i. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

j. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

k. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. In **Sync scope**, for **Sync contents**—Choose to sync from the following content types: Pages, page comments, page attachments, blogs, blog comments, blog attachments, personal spaces, and archived spaces.

> **Note**
>
> Page comments and page attachments can only be seleted if you choose to sync **Pages**. Blog comments and blog attachments can only be seleted if you choose to sync **Blogs**.

> **Important**
>
> If you don't specify a space key regex pattern in **Additional configuration**, all pages and blogs will be crawled by default.

b. In **Additional configuration**, for **Maximum file size**—Specify the file size limit in MBs that Amazon Kendra will crawl. Amazon Kendra will crawl only the files within the size limit you define. The default file size is 50 MB. The maximum file size should be greater than 0 MB and less than or equal to 50 MB.

   For **Spaces regex patterns**—Specify whether to include or exclude specific spaces in your index using:

   - Space key (for example, *my-space-123*)

     > **Note**
     >
     > If you don't specify a space key regex pattern, all pages and blogs will be crawled by default.

   - URL (for example, *.*/MySite/MyDocuments/*)
   - File type (for example, *.*\.pdf, .*\.txt*)

   For **Entity title regex patterns**—Specify regular expression patterns to include or exclude certain blogs, pages, comments, and attachments by titles.

> **Note**
>
> If you want to include or exclude crawling a specific page or subpage, you can use page title regex patterns.

   c. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

      - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
      - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

   d. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

   e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. Select from the Amazon Kendra generated default data source fields you want to map to your index. To add custom data source fields, create an index field name to map to and the field data type.

   b. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Confluence

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `CONFLUENCEV2` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Host URL**—Specify the Confluence host URL instance. For example, *`https://example.confluence.com`*.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Authentication type**—Specify the type of authentication, whether `Basic`, `OAuth2`, (Confluence Server only) `Personal-token`.
- (Optional–For Confluence Server only) **SSL certificate location**—Specificy the `S3bucketName` and `s3certificateName` you used to store your SSL certificate.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of a Secrets Manager secret that contains the authentication credentials you configured in Confluence. If you use basic authentication, the secret is stored in a JSON structure with the following keys:

  ```
  {
      "username": "email ID or user name",
      "password": "Confluence API token"
  }
  ```

  If you use OAuth 2.0 authentication, the secret is stored in a JSON structure with the following keys:

  ```
  {
  ```

```
    "confluenceAppKey": "app key",
    "confluenceAppSecret": "app secret",
    "confluenceAccessToken": "access token",
    "confluenceRefreshToken": "refresh token"
}
```

(For Confluence Server only) If you use basic authentication, the secret is stored in a JSON structure with the following keys:

```
{
    "hostUrl": "Confluence Server host URL",
    "username": "Confluence Server user name",
    "password": "Confluence Server password"
}
```

(For Confluence Server only) If you use Personal Access Token authentication, the secret is stored in a JSON structure with the following keys:

```
{
    "hostUrl": "Confluence Server host URL",
    "patToken": "personal access token"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Confluence connector and Amazon Kendra. For more information, see IAM roles for Confluence data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **File size**—Specify the maximun file size to crawl.
- **Document/content types**—Specify whether to crawl pages, page comments, page attachments, blogs, blog comments, blog attachments, spaces and archived spaces.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain spaces, pages, blogs, and their comments and attachments.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Web proxy**—Specify your web proxy information if you want to connect to your Confluence URL instance via a web proxy. You can use this option for Confluence Server.
- **Access control list (ACL)**—Specify whether to crawl ACL information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **Field mappings**—Choose to map your Confluence data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Confluence template schema.

Notes

- Personal Access Token (PAT) is not available for Confluence Cloud.

Confluence connector V1.0

Confluence is a collaborative work-management tool designed for sharing, storing, and working on project planning, software development, and product management. You can use Amazon Kendra to index your Confluence spaces, pages (including nested pages), blogs, and comments and attachments to indexed pages and blogs.

> **Note**
>
> Confluence connector V1.0 / ConfluenceConfiguration API ended in 2023. We recommend migrating to or using Confluence connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra Confluence data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more

Supported features

Amazon Kendra Confluence data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters
- (For Confluence Server only) Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Confluence data source, make these changes in your Confluence and AWS accounts.

In Confluence, make sure you have:

- Granted Amazon Kendra permissions to view all content within your Confluence instance by:
  - Making Amazon Kendra a member of `confluence-administrators` group.
  - Granting site-admin permissions for all existing spaces, blogs, and pages.
- Copied the URL of your Confluence instance.
- **For SSO (Single Sign-On) users:** Activated the **Show on login page** for the user name and password when you configure Confluence **Authentication methods** in Confluence Data Center.
- **For Confluence Server**
  - Noted your basic authentication credentials containing your Confluence administrative account user name and password to connect to Amazon Kendra.

    > **Note**
    >
    > We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

  - **Optional:**Generated a personal access token in your Confluence account to connect to Amazon Kendra. For more information, see [Confluence documentation on generating personal access tokens](#).
- **For Confluence Cloud**
  - Noted your basic authentication credentials containing your Confluence administrative account user name and password to connect to Amazon Kendra.
- Checked each document is unique in Confluence and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Confluence authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Confluence data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Confluence data source, you must provide details of your Confluence credentials so that Amazon Kendra can access your data. If you have not yet configured Confluence for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Confluence

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Confluence connector V1.0**, and then choose **Add data source**.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name—**Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description—**Enter an optional description for your data source.

   c. In **Default language—**Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag—**Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. Choose between **Confluence Cloud** and **Confluence Server**.

   b. If you choose **Confluence Cloud**, enter the following information:

      i. **Confluence URL—**Your Confluence URL.

      ii. **AWS Secrets Manager secret—**Choose an existing secret or create a new Secrets Manager secret to store your Confluence authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

      - Enter following information in the **Create an AWS Secrets Manager secret window**:

        I. **Secret name—**A name for your secret. The prefix 'AmazonKendra-Confluence-' is automatically added to your secret name.

        II. For **User name** and **Password—**Enter your Confluence user name and password.

III. Choose **Save authentication**.

c. If you choose **Confluence Server**, enter the following information:

   i. **Confluence URL**—Your Confluence user name and password.

   ii. (Optional) For **Web proxy** enter the following information:

      A. **Host name**—Host name for your Confluence account.

      B. **Port number**—Port used by the host URL transport protocol.

   iii. For **Authentication**, Choose either **Basic authentication** or (Confluence Server only) **Personal Access Token**.

   iv. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Confluence authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

      - Enter following information in the **Create an AWS Secrets Manager secret window**:

         I. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Confluence-' is automatically added to your secret name.

         II. For **User name** and **Password**—Enter the authentication credential values you configured in Confluence. If using basic authentication, use your Confluence user name (email ID) and password (API token). If using personal access token, enter the details of the **Personal Access Token** you configured in Confluence account.

         III. Save and add your secret.

d. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

e. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. For **Include personal spaces** and **Include archived spaces—**Choose the optional space types to include in this data source.

b. For **Additional configuration—**Specify regular expression patterns to include or exclude certain content. You can add up to 100 patterns.

c. You can also choose to **Crawl attachments within chosen spaces**.

d. In **Sync run schedule**, for **Frequency—**Choose how often Amazon Kendra will sync with your data source.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. For **Space**, **Page**, **Blog—**Select from the Amazon Kendra generated default data source fields or **Additional suggested field mappings** to add index fields.

   b. **Add field—**To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Confluence

You must specify the following using [ConfluenceConfiguration](https://docs.aws.amazon.com/kendra/latest/APIReference/API_ConfluenceConfiguration.html) API:

- **Confluence version—**Specify the version of the Confluence instance you are using as `CLOUD` or `SERVER`.
- **Secret Amazon Resource Name (ARN)—**Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains your Confluence authentication credentials.

  If you use Confluence Server, you can use either your Confluence user name and password, or your personal access token as the authentication credentials.

  If you use your Confluence user name and password as authentication credentials, you store the following credentials as a JSON structure in your Secrets Manager secret:

```
{
    "username": "user name",
    "password": "password"
}
```

If you use a personal access token to connect Confluence Server to Amazon Kendra, you store the following credentials as a JSON structure in your Secrets Manager secret:

```
{
    "patToken": "personal access token"
}
```

If you use Confluence Cloud, you use your Confluence user name and an API token, configured in Confluence, as your password. You store the following credentials as a JSON structure in your Secrets Manager secret:

```
{
    "username": "user name",
    "password": "API token"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Confluence connector and Amazon Kendra. For more information, see [IAM roles for Confluence data sources](#).

You can also add the following optional features:

- **Web proxy**—Whether to connect to your Confluence URL instance via a web proxy. You can use this option for Confluence Server.
- (For Confluence Server only) **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` as part of the data source configuration. See [Configuring Amazon Kendra to use a VPC](#).
- **Inclusion and exclusion filters**—Specify regular expression patterns to include or exclude certain spaces, blog posts, pages, spaces, and attachments. If you choose to index attachments, only attachments to the indexed pages and blogs are indexed.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Field mappings**—Choose to map your Confluence data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

Learn more

To learn more about integrating Amazon Kendra with your Confluence data source, see:

- Configuring your Amazon Kendra Confluence Server connector

Custom data source connector

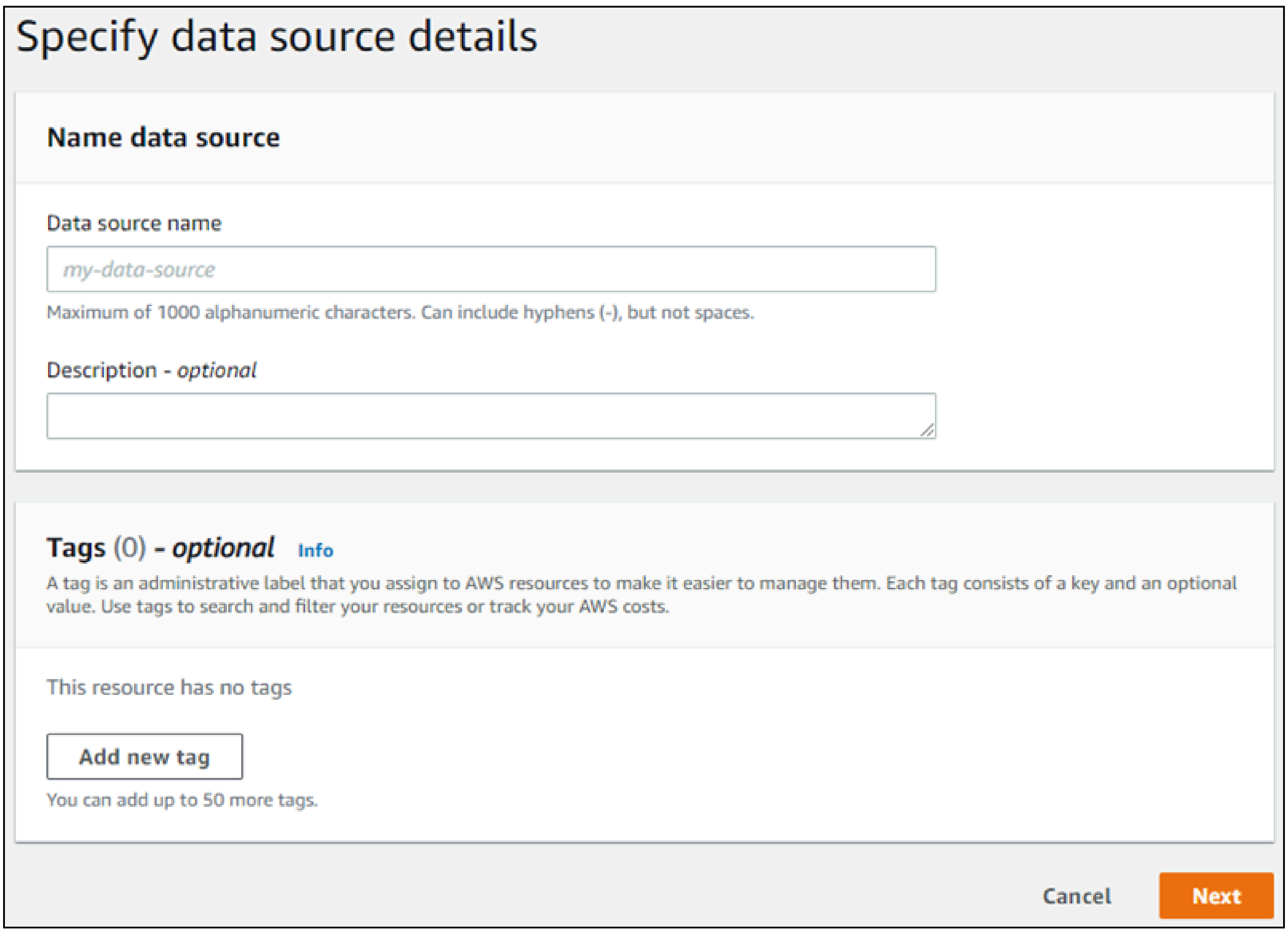
Use a custom data source when you have a repository that Amazon Kendra doesn't yet provide a data source connector for. You can use it to see the same run history metrics that Amazon Kendra data sources provide even when you can't use Amazon Kendra's data sources to sync your repositories. Use this to create a consistent sync monitoring experience between Amazon Kendra data sources and custom ones. Specifically, use a custom data source to see sync metrics for a data source connector that you created using the BatchPutDocument and BatchDeleteDocument APIs.

For troubleshooting your Amazon Kendra custom data source connector, see Troubleshooting data sources.

When you create a custom data source, you have complete control over how the documents to index are selected. Amazon Kendra only provides metric information that you can use to monitor your data source sync jobs. You must create and run the crawler that determines the documents your data source indexes.

You must specify the main title of your documents using the Document object, and `_source_uri` in DocumentAttribute in order to have `DocumentTitle` and `DocumentURI` included in the response of the `Query` result.

You create an identifier for your custom data source using the console or by using the CreateDataSource API. To use the console, give your data source a name, and optionally a description and resource tags. After the data source is created, a data source ID is shown. Copy this ID to use when you synchronize the data source with the index.



You can also create a custom data source using the `CreateDataSource` API. The API returns an ID to use when you synchronize the data source. When you use the `CreateDataSource` API to create a custom data source, you can't set the `Configuration`, `RoleArn` or `Schedule` parameters. If you set these parameters, Amazon Kendra returns a `ValidationException` exception.

To use a custom data source, create an application that is responsible for updating the Amazon Kendra index. The application depends on a crawler that you create. The crawler reads the documents in your repository and determines which should be sent to Amazon Kendra. Your application should perform the following steps:

1. Crawl your repository and make a list of the documents in your repository that are added, updated, or deleted.
2. Call the StartDataSourceSyncJob API to signal that a sync job is starting. You provide a data source ID to identify the data source that is synchronizing. Amazon Kendra returns a execution ID to identify a particular sync job.
3. Call the BatchDeleteDocument API to remove documents from the index. You provide the data source ID and execution ID to identify the data source that is synchronizing and the job that this update is associated with.
4. Call the StopDataSourceSyncJob API to signal the end of the sync job. After you call the `StopDataSourceSyncJob` API, the associated execution ID is no longer valid.
5. Call the ListDataSourceSyncJobs API with the index and data source identifiers to list the sync jobs for the data source and to see metrics for the sync jobs.

After you end a sync job, you can start a new synchronization job. There can be a period of time before all of the submitted documents are added to the index. Use the `ListDataSourceSyncJobs` API to see the status of the sync job. If the `Status` returned for the sync job is `SYNCING_INDEXING`, some documents are still being indexed. You can start a new sync job when the status of the previous job is `FAILED` or `SUCCEEDED`.

After you call the `StopDataSourceSyncJob` API, you can't use a sync job identifier in a call to the `BatchPutDocument` or `BatchDeleteDocument` APIs. If you do, all of the documents submitted are returned in the `FailedDocuments` response message from the API.

Required attributes

When you submit a document to Amazon Kendra using the `BatchPutDocument` API, each document requires two attributes to identify the data source and synchronization run that it belongs to. You must provide the following two attributes to map documents from your custom data source correctly to an Amazon Kendra index:

- `_data_source_id`—The identifier of the data source. This is returned when you create the data source with the console or the `CreateDataSource` API.
- `_data_source_sync_job_execution_id`—The identifier of the sync run. This is returned when you start the index synchronization with the `StartDataSourceSyncJob` API.

The following is the JSON required to index a document using a custom data source.

```
{
    "Documents": [
        {
            "Attributes": [
                {
                    "Key": "_data_source_id",
                    "Value": {
                        "StringValue": "data source identifier"
                    }
                },
                {
                    "Key": "_data_source_sync_job_execution_id",
                    "Value": {
                        "StringValue": "sync job identifier"
                    }
                }
            ],
            "Blob": "document content",
            "ContentType": "content type",
            "Id": "document identifier",
            "Title": "document title"
        }
    ],
    "IndexId": "index identifier",
    "RoleArn": "IAM role ARN"
}
```

When you remove a document from the index using the `BatchDeleteDocument` API, you need to specify the following two fields in the `DataSourceSyncJobMetricTarget` parameter:

- `DataSourceId`—The identifier of the data source. This is returned when you create the data source with the console or the `CreateDataSource` API.
- `DataSourceSyncJobId`—The identifier of the sync run. This is returned when you start the index synchronization with the `StartDataSourceSyncJob` API.

The following is the JSON required to delete a document from the index using the `BatchDeleteDocument` API.

```
{
    "DataSourceSyncJobMetricTarget": {
        "DataSourceId": "data source identifier",
        "DataSourceSyncJobId": "sync job identifier"
    },
    "DocumentIdList": [
        "document identifier"
    ],
    "IndexId": "index identifier"
}
```

Viewing metrics

After a sync job is finished, you can use the DataSourceSyncJobMetrics API to get the metrics associated with the sync job. Use this to monitor your custom data source syncs.

If you submit the same document multiple times, either as part of the `BatchPutDocument` API, the `BatchDeleteDocument` API, or if the document is submitted for both addition and deletion, the document is only counted once in the metrics.

- `DocumentsAdded`—The number of documents submitted using the `BatchPutDocument` API associated with this sync job added to the index for the first time. If a document is submitted for addition more than once in a sync, the document is only counted once in the metrics.
- `DocumentsDeleted`—The number of documents submitted using the `BatchDeleteDocument` API associated with this sync job deleted from the index. If a document is submitted for deletion more than once in a sync, the document is only counted once in the metrics.
- `DocumentsFailed`—The number of documents associated with this sync job that failed indexing. These are documents that were accepted by Amazon Kendra for indexing but could not be indexed or deleted. If a document isn't accepted by Amazon Kendra, the identifier for the

document is returned in the `FailedDocuments` response property of the `BatchPutDocument` and `BatchDeleteDocument` APIs.

- `DocumentsModified`—The number of modified documents submitted using the `BatchPutDocument` API associated with this sync job that were modified in the Amazon Kendra index.

Amazon Kendra also emits Amazon CloudWatch metrics while indexing documents. For more information, see Monitoring Amazon Kendra with Amazon CloudWatch.

Amazon Kendra doesn't return the `DocumentsScanned` metric for custom data sources. It also emits the CloudWatch metrics listed in the document Metrics for Amazon Kendra data sources.

Learn more

To learn more about integrating Amazon Kendra with your custom data source, see:

- Adding custom data sources to Amazon Kendra

Custom data source (Java)

The following code provides a sample implementation of a custom data source using Java. The program first creates a custom data source and then synchronizes newly added documents to the index with the custom data source.

The following code demonstrates creating and using a custom data source. When you use a custom data source in your application you don't need to create a new data source (one-off process) each time that you synchronize your index with your data source. You use the index ID and data source ID to synchronize your data.

```
package com.amazonaws.kendra;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.services.kendra.KendraClient;
import csoftware.amazon.awssdk.services.kendra.model.BatchPutDocumentRequest;
import csoftware.amazon.awssdk.services.kendra.model.BatchPutDocumentResponse;
import software.amazon.awssdk.services.kendra.model.CreateDataSourceRequest;
import software.amazon.awssdk.services.kendra.model.CreateDataSourceResponse;
import software.amazon.awssdk.services.kendra.model.DataSourceType;
import software.amazon.awssdk.services.kendra.model.Document;
import software.amazon.awssdk.services.kendra.model.ListDataSourceSyncJobsRequest;
```

```
import software.amazon.awssdk.services.kendra.model.ListDataSourceSyncJobsResponse;
import software.amazon.awssdk.services.kendra.model.StartDataSourceSyncJobRequest;
import software.amazon.awssdk.services.kendra.model.StartDataSourceSyncJobResponse;
import software.amazon.awssdk.services.kendra.model.StopDataSourceSyncJobRequest;
import software.amazon.awssdk.services.kendra.model.StopDataSourceSyncJobResponse;

public class SampleSyncForCustomDataSource {
  public static void main(String[] args) {
    KendraClient kendra = KendraClient.builder().build();

    String myIndexId = "yourIndexId";
    String dataSourceName = "custom data source";
    String dataSourceDescription = "Amazon Kendra custom data source connector"

    // Create custom data source
    CreateDataSourceRequest createDataSourceRequest = CreateDataSourceRequest
        .builder()
        .indexId(myIndexId)
        .name(dataSourceName)
        .description(dataSourceDescription)
        .type(DataSourceType.CUSTOM)
        .build();

    CreateDataSourceResponse createDataSourceResponse =
 kendra.createDataSource(createDataSourceRequest);
    System.out.println(String.format("Response of creating data source: %s",
 createDataSourceResponse));

    // Get the data source ID from createDataSourceResponse
    String dataSourceId = createDataSourceResponse.Id();

    // Wait for the custom data source to become active
    System.out.println(String.format("Waiting for Amazon Kendra to create the data
 source %s", dataSourceId));
    // You can use the DescribeDataSource API to check the status
    DescribeDataSourceRequest describeDataSourceRequest = DescribeDataSourceRequest
        .builder()
        .indexId(myIndexId)
        .id(dataSourceId)
        .build();

    while (true) {
        DescribeDataSourceResponse describeDataSourceResponse =
 kendra.describeDataSource(describeDataSourceRequest);
```

```
        DataSourceStatus status = describeDataSourceResponse.status();
        System.out.println(String.format("Creating data source. Status: %s", status));
        if (status != DataSourceStatus.CREATING) {
            break;
        }

        TimeUnit.SECONDS.sleep(60);
    }

    // Start syncing yor data source by calling StartDataSourceSyncJob and providing
your index ID
    // and your custom data source ID
    System.out.println(String.format("Synchronize the data source %s", dataSourceId));
    StartDataSourceSyncJobRequest startDataSourceSyncJobRequest =
StartDataSourceSyncJobRequest
        .builder()
        .indexId(myIndexId)
        .id(dataSourceId)
        .build();
    StartDataSourceSyncJobResponse startDataSourceSyncJobResponse =
kendra.startDataSourceSyncJob(startDataSourceSyncJobRequest);

    // Get the  sync job execution ID from startDataSourceSyncJobResponse
    String executionId = startDataSourceSyncJobResponse.ExecutionId();
System.out.println(String.format("Waiting for the data source to sync with the index
%s for execution ID %s", indexId, startDataSourceSyncJobResponse.executionId()));

    // Add 2 documents uploaded to S3 bucket to your index using the BatchPutDocument
API
    // The added documents should sync with your custom data source
    Document pollyDoc = Document
        .builder()
        .s3Path(
            S3Path.builder()
            .bucket("amzn-s3-demo-bucket")
            .key("what_is_Amazon_Polly.docx")
            .build())
        .title("What is Amazon Polly?")
        .id("polly_doc_1")
        .build();

    Document rekognitionDoc = Document
        .builder()
```

```
        .s3Path(
            S3Path.builder()
            .bucket("amzn-s3-demo-bucket")
            .key("what_is_amazon_rekognition.docx")
            .build())
        .title("What is Amazon rekognition?")
        .id("rekognition_doc_1")
        .build();

    BatchPutDocumentRequest batchPutDocumentRequest = BatchPutDocumentRequest
        .builder()
        .indexId(myIndexId)
        .documents(pollyDoc, rekognitionDoc)
        .build();

    BatchPutDocumentResponse result = kendra.batchPutDocument(batchPutDocumentRequest);
    System.out.println(String.format("BatchPutDocument result: %s", result));

    // Once custom data source synced, stop the sync job using the
 StopDataSourceSyncJob API
    StopDataSourceSyncJobResponse stopDataSourceSyncJobResponse =
 kendra.stopDataSourceSyncJob(
        StopDataSourceSyncJobRequest()
            .indexId(myIndexId)
            .id(dataSourceId)
    );

 // List your sync jobs
    ListDataSourceSyncJobsRequest listDataSourceSyncJobsRequest =
 ListDataSourceSyncJobsRequest
        .builder()
        .indexId(myIndexId)
        .id(dataSourceId)
        .build();

    while (true) {
        ListDataSourceSyncJobsResponse listDataSourceSyncJobsResponse =
 kendra.listDataSourceSyncJobs(listDataSourceSyncJobsRequest);
        DataSourceSyncJob job = listDataSourceSyncJobsResponse.history().get(0);
        System.out.println(String.format("Status: %s", job.status()));
    }
  }
}
```

Dropbox

Dropbox is a file hosting service that offers cloud storage, document organization, and document templating services. If you are a Dropbox user, you can use Amazon Kendra to index your Dropbox files, Dropbox Paper, Dropbox Paper Templates, and stored shortcuts to web pages. You can also configure Amazon Kendra to index specific Dropbox files, Dropbox Paper, Dropbox Paper Templates, and stored shortcuts to web pages.

Amazon Kendra supports both Dropbox and Dropbox Advanced for Dropbox Business.

You can connect Amazon Kendra to your Dropbox data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Dropbox data source connector, see Troubleshooting data sources.

Topics

Supported features

Amazon Kendra Dropbox data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Dropbox data source, make these changes in your Dropbox and AWS accounts.

In Dropbox, make sure you have:

- Created a Dropbox Advanced account and set up an admin user.
- Configured a Dropbox app with a unique **App name**, activated **Scoped Access**. See Dropbox documentation on creating an app.
- Activated **Full Dropbox** permissions on the Dropbox console and added the following permissions:
  - files.content.read
  - files.metadata.read
  - sharing.read
  - file_requests.read
  - groups.read
  - team_info.read
  - team_data.content.read
- Noted your Dropbox app key, Dropbox app secret, and Dropbox access token for basic authentication credentials.

  > **Note**
  >
  > We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Configured and copied a temporary Oauth 2.0 access token for your Dropbox app. This token is temporary and expires after 4 hours. See Dropbox documentation on OAuth authentication.

  > **Note**
  >
  > It is recommended that you create a Dropbox refresh access token that never expires, rather that relying on a one-time access token that expires after 4 hours. A refresh access token is permanent and never expires so that you can continue to sync your data source in the future.

- **Recommended:** Configured a Dropbox permanent refresh token that never expires to allow Amazon Kendra to continue to sync your data source without any disruptions. See Dropbox documentation on refresh tokens.
- Checked each document is unique in Dropbox and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Dropbox authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don’t have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Dropbox data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Dropbox data source, you must provide the necessary details of your Dropbox data source so that Amazon Kendra can access your data. If you have not yet configured Dropbox for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to Dropbox

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Dropbox connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Dropbox connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

b. **Type of authentication token**—Choose either a permanent token (recommended) or a temporary access token.

c. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Dropbox authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

    i. Enter following information in the **Create an AWS Secrets Manager secret window**:

        A. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Dropbox-' is automatically added to your secret name.

        B. For **App key**, **App secret**, and token information (permanent or temporary)—Enter the authentication credential values configured in Dropbox.

    ii. Save and add your secret.

d. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

e. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

f. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

   g. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. For **Select entities or content types**—Choose Dropbox entities or content types you want to crawl.

   b. In **Additional configuration** for **Regex patterns**—Add regular expression patterns to include or exclude certain files.

   c. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

      - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
      - New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
      - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

   d. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

   e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. **Files**, **Dropbox Paper**, and **Dropbox Paper templates**—Select from the Amazon Kendra generated default data source fields you want to map to your index.

b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Dropbox

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `DROPBOX` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Access token type**—Specify whether you want to use permanent or temporary access token for your AWS Secrets Manager secret that stores your authentication crednetials.

> **Note**
>
> It's recommended that you create a refresh access token that never expires in Dropbox rather that relying on a one-time access token that expires after 4 hours. You create an app and a refresh access token in the Dropbox developer console and provide the access token in your secret.

- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Dropbox account. The secret is stored in a JSON structure with the following keys:

```
{
    "appKey": "Dropbox app key",
    "appSecret": "Dropbox app secret",
    "accesstoken": "temporary access token or refresh access token"
}
```

- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Dropbox connector and Amazon Kendra. For more information, see IAM roles for Dropbox data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.

- **Document/content types**—Specify whether to crawl files in your Dropbox, Dropbox Paper documents, Dropbox Paper templates, and web page shortcuts stored in your Dropbox.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain files.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Access control list (ACL)**—Specify whether to crawl ACL information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Dropbox data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Dropbox template schema.

Learn more

To learn more about integrating Amazon Kendra with your Dropbox data source, see:

- Index your Dropbox content using the Dropbox connector for Amazon Kendra

Notes

- When Access Control Lists (ACLs) are enabled, the "Sync only new or modified content" option is not available due to Dropbox API limitations. We recommend using "Full sync" or "New, modified, or deleted content sync" modes instead, or disable ACLs if you need to use this sync mode.

Drupal

> **Note**
>
> Drupal connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

Drupal is an open-source content management system (CMS) that you can use to create websites and web applications. You can use Amazon Kendra to index the following in Drupal:

- Content—Articles, Basic pages, Basic blocks, User defined content types, User defined block types, Custom content types, Custom block types
- Comment—For any Content type and Block type
- Attachments—For any Content type and Block type

You can connect Amazon Kendra to your Drupal data source using the Amazon Kendra console or the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Drupal data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions

- Notes

Supported features

Amazon Kendra Drupal data source connector supports the following features:

- Field mappings
- User context filtering
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Drupal data source, make these changes in your Drupal and AWS accounts.

In Drupal, make sure you have:

- Created a Drupal (Standard) Suite account and a user with an administrator role.
- Copied your Drupal site name and configured a host url. For example, *`https://<hostname>/<drupalsitename>`*.
- Configured basic authentication credentials containing a user name (Drupal website login user name) and password (Drupal website password).
- **Recommended:** Configured an OAuth 2.0 credential token. Use this token along with your Drupal password grant, client id, client secret, user name (Drupal website login user name) and password (Drupal website password) to connect to Amazon Kendra.
- Added the following permissions in your Drupal account using an administrator role:
  - administer blocks
  - administer block_content display
  - administer block_content fields
  - administer block_content form display
  - administer views
  - view user email addresses

- view own unpublished content
- view page revisions
- view article revisions
- view all revisions
- view the administration theme
- access content
- access content overview
- access comments
- search content
- access files overview
- access contextual links

> **Note**
>
> If there are user defined content types or user defined block types, or any views and blocks are added to the Drupal website, they must be provided with administrator access.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Drupal authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you

re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Drupal data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Drupal data source you must provide details of your Drupal credentials so that Amazon Kendra can access your data. If you have not yet configured Drupal for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Drupal

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Drupal connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Drupal connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. In **Source**, for **Host URL**—The host URL of your Drupal site. For example, *https://<hostname>/<drupalsitename>*.

   b. For **SSL certificate location**—Enter the path to the SSL certificate stored in your Amazon S3 bucket.

   c. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

   d. For **Authentication**—Choose between **Basic authentication** and **OAuth 2.0 authentication** based on your use case.

   e. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Drupal authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

      i. Enter following information in the **Create an AWS Secrets Manager secret window**:

         A. If you chose **Basic authentication**, enter a **Secret Name**, the **User name**, (Drupal site user name), and **Password** (Drupal site password) that you copied and choose **Save and add secret**.

         B. If you chose **OAuth 2.0 authentication**, enter a **Secret Name**, **User name** (Drupal site user name), **Password** (Drupal site password), **Client ID**, and **Client secret** generated in your Drupal account and choose **Save and add secret**.

      ii. Choose **Save**.

f. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

g. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. For **Sync scope**, choose from the following options:

> **Note**
>
> When you choose to crawl **Articles**, **Basic pages**, and **Basic blocks**, their default fields will be synced automatically. You can also choose to sync their comments, attachments, custom fields and other custom entities.

- For **Select entities**:

  - **Articles**—Choose whether to crawl **Articles**, their comments **Comments**, and their **Attachments**.

        - **Basic pages—**Choose whether to crawl **Basic pages**, their **Comments**, and their **Attachments**.
        - **Basic blocks—**Choose whether to crawl **Basic blocks**, their **Comments**, and their **Attachments**.
        - You can also choose to add **Custom content types** and **Custom Blocks**.

    b. For **Additional configuration – optional**:

        - For **Regex pattern—**Add regular expression patterns to include or exclude specific entity titles and file names. You can add up to 100 patterns.

    c. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

        - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
        - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

    d. In **Sync run schedule**, **Frequency—**How often Amazon Kendra will sync with your data source.

    e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

    a. For **Contents**, **Comments**, and **Attachments—**Select from the Amazon Kendra generated default data source fields you want to map to your index.

    b. **Add field—**To add custom data source fields to create an index field name to map to and the field data type.

    c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Drupal

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source—**Specify the data source type as `DRUPAL` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Sync mode—**Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)—**Provide the Amazon Resource Name (ARN) of a Secrets Manager secret that contains the authentication credentials you created in your Drupal account.

  If you use basic authentication, the secret is stored in a JSON structure with the following keys:

  ```
  {
      "username": "user name",
      "password": "password"
  }
  ```

  If you use OAuth 2.0 authentication, the secret is stored in a JSON structure with the following keys:

```
{
    "username": "user name",
    "password": "password",
    "clientId": "client id",
    "clientSecret": "client secret"
}
```

> **Note**
>
> > **Note**
> >
> > We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Drupal connector and Amazon Kendra. For more information, see IAM roles for Drupal data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include contents, comments, and attachments. You can also specify regular expression patterns to include or exclude contents, comments, and attachments.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the

inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **Field mappings**—Choose to map your Drupal data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Drupal template schema.

Notes

- Drupal APIs have no official throttling limits.
- Java SDKs are not available for Drupal.
- Drupal data can be fetched only using native JSON API's.
- Content types not associated with any Drupal **View** cannot be crawled.
- You need administrator access to crawl data from Drupal **Blocks**.
- There is no JSON API available to create the user defined content type using HTTP verbs.
- The document body and comments for **Articles**, **Basic pages**, **Basic blocks**, user defined content type, and user defined block type, are displayed in HTML format. If the HTML content is not well-

formed, then the HTML related tags will appear in the document body and comments and will be visible in Amazon Kendra search results.

- Content types and **Block** types without description or body will not be ingested into Amazon Kendra. Only **Comments** and **Attachments** of such **Content** or **Block** types will be ingested into your Amazon Kendra index.

GitHub

GitHub is a web-based hosting service for software development providing code storage and management services with version control. You can use Amazon Kendra to index your GitHub Enterprise Cloud (SaaS) and GitHub Enterprise Server (On Prem) repository files, issue and pull requests, issue and pull request comments, and issue and pull request comment attachments. You can also choose to include or exclude certain files.

> **Note**
>
> Amazon Kendra now supports an upgraded GitHub connector.
> The console has been automatically upgraded for you. Any new connectors you create in the console will use the upgraded architecture. If you use the API, you must now use the TemplateConfiguration object instead of the `GitHubConfiguration` object to configure your connector.
> Connectors configured using the older console and API architecture will continue to function as configured. However, you won't be able to edit or update them. If you want to edit or update your connector configuration, you must create a new connector.
> We recommended migrating your connector workflow to the upgraded version. Support for connectors configured using the older architecture is scheduled to end by June 2024.

You can connect Amazon Kendra to your GitHub data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra GitHub data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites

- Connection instructions
- Learn more

Supported features

Amazon Kendra GitHub data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your GitHub data source, make these changes in your GitHub and AWS accounts.

In GitHub, make sure you have:

- Created a GitHub user with administrative permissions to the GitHub organization.
- Configured a personal access token in Git Hub to use as your authentication credentials. See GitHub documentation on creating a personal access token.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **Recommended:**Configured an OAuth token for authentication credentials. Use OAuth token for better API throttle limits and connector performance. See GitHub documentation on OAuth authorization.
- Noted the GitHub host URL for the type of GitHub service that you use. For example, the host URL for GitHub cloud could be *`https://api.github.com`* and the host URL for GitHub server could be *`https://on-prem-host-url/api/v3/`*.

- Noted the name of your organization for GitHub the GitHub Enterprise Cloud (SaaS) account or GitHub Enterprise Server (on-premises) account you want to connect to. You can find your organization name by logging into GitHub desktop and selecting **Your organizations** under your profile picture dropdown.
- **Optional (server only):** Generated a SSL certificate and copied the path to the certificate stored in an Amazon S3 bucket. You use this to connect to GitHub if you require a secure SSL connection. You can simply generate a self-signed X509 certificate on any computer using OpenSSL. For an example of using OpenSSL to create an X509 certificate, see [Create and sign an X509 certificate](https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/configuring-https-ssl.html).
- Added the following permissions:

  For GitHub Enterprise Cloud (SaaS)

  - `repo:status` – Grants read/write access to commit statuses in public and private repositories. This scope is only necessary to grant other users or services access to private repository commit statuses without granting access to the code.
  - `repo_deployment` – Grants access to deployment statuses for public and private repositories. This scope is only necessary to grant other users or services access to deployment statuses, without granting access to the code.
  - `public_repo` – Limits access to public repositories. That includes read/write access to code, commit statuses, repository projects, collaborators, and deployment statuses for public repositories and organizations. Also required for starring public repositories.
  - `repo:invite` – Grants accept/decline abilities for invitations to collaborate on a repository. This scope is only necessary to grant other users or services access to invites without granting access to the code.
  - `security_events` – Grants: read and write access to security events in the code scanning API. This scope is only necessary to grant other users or services access to security events without granting access to the code.
  - `read:org` – Read-only access to organization membership, organization projects, and team membership.
  - `user:email` – Grants read access to a user's email addresses. Required by Amazon Kendra to crawl ACLs.
  - `user:follow` – Grants access to follow or unfollow other users. Required by Amazon Kendra to crawl ACLs.

- `read:user` – Grants access to read a user's profile data. Required by Amazon Kendra to crawl ACLs.
- `workflow` – Grants the ability to add and update GitHub Actions workflow files. Workflow files can be committed without this scope if the same file (with both the same path and contents) exists on another branch in the same repository.

For more information, see Scopes for OAuth apps in GitHub Docs.

For GitHub Enterprise Server (On Prem)

- `repo:status` – Grants read/write access to commit statuses in public and private repositories. This scope is only necessary to grant other users or services access to private repository commit statuses without granting access to the code.
- `repo_deployment` – Grants access to deployment statuses for public and private repositories. This scope is only necessary to grant other users or services access to deployment statuses, without granting access to the code.
- `public_repo` – Limits access to public repositories. That includes read/write access to code, commit statuses, repository projects, collaborators, and deployment statuses for public repositories and organizations. Also required for starring public repositories.
- `repo:invite` – Grants accept/decline abilities for invitations to collaborate on a repository. This scope is only necessary to grant other users or services access to invites without granting access to the code.
- `security_events` – Grants: read and write access to security events in the code scanning API. This scope is only necessary to grant other users or services access to security events without granting access to the code.
- `read:user` – Grants access to read a user's profile data. Required by Amazon Q Business to crawl ACLs.
- `user:email` – Grants read access to a user's email addresses. Required by Amazon Q Business to crawl ACLs.
- `user:follow` – Grants access to follow or unfollow other users. Required by Amazon Q Business to crawl ACLs.
- `site_admin` – Grants site administrators access to GitHub Enterprise Server Administration API endpoints.
- `workflow` – Grants the ability to add and update GitHub Actions workflow files. Workflow files can be committed without this scope if the same file (with both the same path and contents) exists on another branch in the same repository.

For more information, see Scopes for OAuth apps in GitHub Docs and Understanding scopes for OAuth Apps in GitHub Developer.

- Checked each document is unique in GitHub and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your GitHub authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your GitHub data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your GitHub data source, you must provide the necessary details of your GitHub data source so that Amazon Kendra can access your data. If you have not yet configured GitHub for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to GitHub

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **GitHub connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **GitHub connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. **GitHub source**—Choose between **GitHub Enterprise Cloud** and **GitHub Enterprise Server**.

b. **GitHub host URL**—For example, the host URL for GitHub cloud could be *https://api.github.com* and the host URL for GitHub server could be *https://on-prem-host-url/api/v3/*.

c. **GitHub organization name—**Enter your GitHub organization name. You can find your organization information in your GitHub account.

> **Note**
>
> GitHub connector supports crawling a single organization per data source connector instance.

d. **Authorization—**Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

e. **AWS Secrets Manager secret—**Choose an existing secret or create a new Secrets Manager secret to store your GitHub authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

   i. Enter following information in the **Create an AWS Secrets Manager secret window**:

      A. **Secret name**—A name for your secret. The prefix 'AmazonKendra-GitHub-' is automatically added to your secret name.

      B. For **GitHub token—**Enter the authentication credential value configured in GitHub.

   ii. Save and add your secret.

f. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

g. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. **Select repositories**—Choose to crawl all repositories or select.

      If you choose to crawl select repositories, add the names for the repositories and, optionally, the name of any specific branches.

   b. **Content types**—Choose the content types you want to crawl from files, issues, pull requests, and more.

   c. **Regex patterns**—Add regular expression patterns to include or exclude certain files.

   d. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

      - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
      - New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
      - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

e. In **Sync run schedule** for **Frequency**—Choose how often to sync your data source content and update your index.

f. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. **Default data source fields**—Select from the Amazon Kendra generated default data source fields you want to map to your index.

   b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to GitHub

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `GITHUB` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **GitHub type**—Specify the type as either `SAAS` or `ON_PREMISE`.
- **Host URL**—Specify the GitHub host URL or API endpoint URL. For example, if you use GitHub SaaS/Enterprise Cloud, the host URL could be `https://api.github.com`, and for GitHub on-premises/Enterprise Server the host URL could be `https://on-prem-host-url/api/v3/`.
- **Organization name**—Specify the name of the organization of the GitHub account. You can find your organization name by logging into GitHub desktop and selecting **Your organizations** under your profile picture dropdown.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all

content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:

  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your GitHub account. The secret is stored in a JSON structure with the following keys:

```
{
    "personalToken": "token"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the GitHub connector and Amazon Kendra. For more information, see IAM roles for GitHub data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see [Configuring Amazon Kendra to use an Amazon VPC](#).

  > **Note**
  >
  > If you use GitHub server, you must use an Amazon VPC to connect to your GitHub server.

- **Repository filter**—Filter repositories by their name and branch names.
- **Document/content types**—Specify whether to crawl repository documents, issues, issue comments, issue comment attachments, pull requests, pull request comments, pull request comment attachments.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain files and folders.

  > **Note**
  >
  > Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Access control list (ACL)**—Specify whether to crawl ACL information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see [User context filtering](#).
- **Field mappings**—Choose to map your GitHub data source fields to your Amazon Kendra index fields. You can include fields of documents, commits, issues, issue attachments, issue comments, pull requests, pull request attachments, pull request comments. For more information, see [Mapping data source fields](#).

  > **Note**
  >
  > The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must

map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see GitHub template schema.

Learn more

To learn more about integrating Amazon Kendra with your GitHub data source, see:

- Reimagine search on GitHub repositories with the power of the Amazon Kendra GitHub connector

Gmail

Gmail is email client developed by Google through which you can send email messages with file attachments. Gmail messages can be sorted and stored inside your email inbox using folders and labels. You can use Amazon Kendra to index your email messages and message attachments. You can also configure Amazon Kendra to include or exclude specific email messages, message attachments, and labels for indexing.

You can connect Amazon Kendra to your Gmail data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Gmail data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more
- Notes

Supported features

- Field mappings

- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Gmail data source, make these changes in your Gmail and AWS accounts.

In Gmail, make sure you have:

- Created a Google Cloud Platform admin account and have created a Google Cloud project.
- Activated Gmail API and Admin SDK API in your admin account.
- Created a service account and downloaded a JSON private key for your Gmail. For information on how to create and access your private key, see Google Cloud documentation on how to [Create a service account key]() and [Service account credentials]().
- Copied your admin account email, your service account email, and your private key to use as your authentication credentials.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Added the following Oauth scopes (using an admin role) for your user and the shared directories you want to index:
  - https://www.googleapis.com/auth/admin.directory.user.readonly
  - https://www.googleapis.com/auth/gmail.readonly
- Checked each document is unique in Gmail and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Gmail authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Gmail data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Gmail data source you must provide details of your Gmail credentials so that Amazon Kendra can access your data. If you have not yet configured Gmail for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to Gmail

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Gmail connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Gmail connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see [User context filtering](#).

   b. In **Authentication** for **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Gmail authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

      - Enter following information in the **Create an AWS Secrets Manager secret window**:

        A. **Secret Name**—A name for your secret.

B. **Client email**—The client email that you copied from your Google service account.

C. **Admin account email**—The admin account email that you would like to use.

D. **Private key**—The private key you copied from your Google service account.

E. Save and add your secret.

c. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

d. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

e. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. For**Entity types**—Choose to sync message attachments.

b. (Optional) For **Additional configuration**, enter the following information:

i. **Date range**—Enter a date range to specify the start and end date of emails you want to crawl.

ii. **Email domains**—Include or exclude certain emails based on "to", "from", "cc" and "bcc" email domains.

iii. **Keywords in subjects**—Include or exclude emails based on keywords in their email subjects.

> **Note**
>
> You can also choose to include any documents that match all the subject keywords you have entered.

iv. **Labels**—Add regular expression patterns to include or exclude certain email labels.

v. **Attachments**—Add regular expression patterns to include or exclude certain email attachments.

c. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

> **Important**
>
> Because there is no API to update permanently deleted Gmail messages, new, modified, or deleted content sync:
>
> - Won't remove messages that were permanently deleted from Gmail from your Amazon Kendra index
> - Won't sync changes in Gmail email labels
>
> To sync your Gmail data source label changes and permanently deleted email messages to your Amazon Kendra index, you must run full crawls periodically.

d. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. **Default data source fields**—Select from the Amazon Kendra generated default data source fields you want to map to your index.

> **Note**
>
> Amazon Kendra Gmail data source connector does not support creating custom index fields due to API limitations.

b. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Gmail

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `GMAIL` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source’s mechanism for tracking content changes and index content that changed since the last sync.

> **Important**
>
> Because there is no API to update permanently deleted Gmail messages, new, modified, or deleted content sync:

  - Won't remove messages that were permanently deleted from Gmail from your Amazon Kendra index
  - Won't sync changes in Gmail email labels

  To sync your Gmail data source label changes and permanently deleted email messages to your Amazon Kendra index, you must run full crawls periodically.

- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Gmail account. The secret is stored in a JSON structure with the following keys:

```
{
    "adminAccountEmailId": "service account email",
    "clientEmailId": "user account email",
    "privateKey": "private key"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Gmail connector and Amazon Kendra. For more information, see IAM roles for Gmail data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain "to", "from", "cc", "bcc" emails.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Gmail data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

> **Note**
>
> Amazon Kendra Gmail data source connector does not support creating custom index fields due to API limitations.

For a list of other important JSON keys to configure, see Gmail template schema.

Learn more

To learn more about integrating Amazon Kendra with your Gmail data source, see:

- Perform intelligent search across emails in your Google workspace using the Gmail connector for Amazon Kendra.

Notes

- Because there is no API to update permanently deleted Gmail messages, a `FULL_CRAWL`/**New, modified, or deleted content sync**:
  - Won't remove messages that were permanently deleted from Gmail from your Amazon Kendra index

  - Won't sync changes in Gmail email labels

  To sync your Gmail data source label changes and permanently deleted email messages to your Amazon Kendra index, you must run full crawls periodically.

- Amazon Kendra Gmail data source connector does not support creating custom index fields due to API limitations.

Google Drive

Google Drive is a cloud-based file storage service. You can use Amazon Kendra to index documents stored in shared drives, My Drives, and Shared with me folders in your Google Drive data source. You can index both Google Workspace documents as well as documents listed in Types of documentation. You can also use inclusion and exclusion filters to index content by file name, file type, and file path.

You can connect Amazon Kendra to your Google Drive data source using the Amazon Kendra console, the TemplateConfiguration API, or the GoogleDriveConfiguration API.

Amazon Kendra has two versions of the Google Drive connector. Supported features of each version include:

Google Drive connector V1.0 / GoogleDriveConfiguration API

- Field mappings
- User access control
- Inclusion/exclusion filters

Google Drive connector V2.0 / TemplateConfiguration API

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

> **Note**
>
> Google Drive connector V1.0 / Google DriveConfiguration API ended in 2023. We recommend migrating to or using Google Drive connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra Google Drive data source connector, see Troubleshooting data sources.

Topics

- Google Drive connector V1.0
- Google Drive connector V2.0

Google Drive connector V1.0

Google Drive is a cloud-based file storage service. You can use Amazon Kendra to index documents and comments stored in shared drives, My Drives, and Shared with me folders in your Google Drive data source. You can index Google Workspace documents, as well as documents listed in Types of documentation. You can also use inclusion and exclusion filters to index content by file name, file type, and file path.

> **Note**
>
> Google Drive connector V1.0 / Google DriveConfiguration API ended in 2023. We recommend migrating to or using Google Drive connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra Google Drive data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more

Supported features

- Field mappings
- User access control
- Inclusion/exclusion filters

Prerequisites

Before you can use Amazon Kendra to index your Google Drive data source, make these changes in your Google Drive and AWS accounts.

In Google Drive, make sure you have:

- **Either** been granted access by a super admin role **or** are a user with administrative privileges. You do not need a super admin role for yourself if you have been granted access by a super admin role.
- Created a service account with **Enable G Suite Domain-wide Delegation** activated and a JSON key as private key using the account.
- Copied your user account email and your service account email. When you connect to Amazon Kendra you enter your user account email as admin account email and your service account email as client email in your AWS Secrets Manager secret.

  > **Note**
  >
  > We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Added Admin SDK API and Google Drive API in your account.
- Added (or asked a user with a super admin role to add) the following permissions to your service account using a super admin role:
  - https://www.googleapis.com/auth/drive.readonly
  - https://www.googleapis.com/auth/drive.metadata.readonly
  - https://www.googleapis.com/auth/admin.directory.user.readonly
  - https://www.googleapis.com/auth/admin.directory.group.readonly

- Checked each document is unique in Google Drive and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Google Drive authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Google Drive data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Google Drive data source, you must provide the necessary details of your Google Drive data source so that Amazon Kendra can access your data. If you have not yet configured Google Drive for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Google Drive

1. Sign in to the AWS Management Console and open the [Amazon Kendra console](https://console.aws.amazon.com/kendra/).
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Google Drive connector V1.0** , and then choose **Add connector**.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. For **Type of authentication**—Choose between **Existing** and **New**. If you choose to use an existing secret, use **Select secret** to choose your secret.

   b. If you choose to create a new secret an AWS Secrets Manager secret option opens.

      - Enter following information in the **Create an AWS Secrets Manager secret window**:

A. **Secret name**—A name for your secret. The prefix ‘AmazonKendra-Google Drive-’ is automatically added to your secret name.

B. For **Admin account email**, **Client email**, and **Private key**—Enter the authentication credential values you generated and downloaded from your Google Drive account.

C. Choose **Save authentication**.

c. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

d. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. **Exclude user accounts**—The Google Drive users you want to exclude from the index. You can add up to 100 user accounts.

   b. **Exclude shared drives**—The Google Drive shared drives you want to exclude from your index. You can add up to 100 shared drives.

   c. **Exclude file types drives**—The Google Drive file types you want to exclude from your index. You can also choose to edit MIME type selections.

   d. **Additional configurations**—Regular expression patterns to include or exclude certain content. You can add up to 100 patterns.

   e. **Frequency**—How often Amazon Kendra will sync with your data source.

   f. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. For **GoogleDrive field name** and **Additional suggested field mappings**—Select from the Amazon Kendra generated default data source fields you want to map to your index.

   b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Google Drive

You must specify the following using the GoogleDriveConfiguration API:

- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Google Drive account. The secret is stored in a JSON structure with the following keys:

```
{
    "clientAccount": "service account email",
    "adminAccount": "user account email"",
    "privateKey": "private key"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Google Drive connector and Amazon Kendra. For more information, see IAM roles for Google Drive data sources.

You can also add the following optional features:

- **Inclusion and exclusion filters**—By default Amazon Kendra indexes all documents in Google Drive. You can specify whether to include or exclude certain content in shared drives, user accounts, document MIME types, and files. If you choose to exclude user accounts, none of the files in the My Drive owned by the account are indexed. Files shared with the user are indexed unless the owner of the file is also excluded.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that

matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Field mappings**—Choose to map your Google Drive data source fields to your Amazon Kendra index fields. For more information, see [Mapping data source fields].

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see [User context filtering].

Learn more

To learn more about integrating Amazon Kendra with your Google Drive data source, see:

- [Getting started with the Amazon Kendra Google Drive connector]

Google Drive connector V2.0

Google Drive is a cloud-based file storage service. You can use Amazon Kendra to index documents and comments stored in shared drives, My Drives, and Shared with me folders in your Google Drive data source. You can index Google Workspace documents, as well as documents listed in [Types of documentation]. You can also use inclusion and exclusion filters to index content by file name, file type, and file path.

> **Note**
>
> Google Drive connector V1.0 / Google DriveConfiguration API ended in 2023. We recommend migrating to or using Google Drive connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra Google Drive data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Notes

Supported features

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Google Drive data source, make these changes in your Google Drive and AWS accounts.

In Google Drive, make sure you have:

- **Either** been granted access by a super admin role **or** are a user with administrative privileges. You do not need a super admin role for yourself if you have been granted access by a super admin role.

- Configured Google Drive Service Account connection credentials containing your admin account email, client email (service account email), and private key. See [Google Cloud documentation on creating and deleting service account keys](#).

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Created a Google Cloud Service Account (an account with delegated authority to assume a user identity) with **Enable G Suite Domain-wide Delegation** activated for server-to-server authentication, and then generated a JSON private key using the account.

> **Note**
>
> The private key should be generated after the creation of the service account.

- Added Admin SDK API and Google Drive API in your user account.
- **Optional:** Configured Google Drive OAuth 2.0 connection credentials containing client ID, client secret, and refresh token as connection credentials for a specific user. You need this to crawl individual account data. See [Google documentation on using OAuth 2.0 to access APIs](#).
- Added (or asked a user with a super admin role to add) the following OAuth scopes to your service account using a super admin role. These API scopes are needed to crawl all documents, and access control (ACL) information for all users in a Google Workspace domain:
  - https://www.googleapis.com/auth/drive.readonly—View and download all your Google Drive files
  - https://www.googleapis.com/auth/drive.metadata.readonly—View metadata for files in your Google Drive
  - https://www.googleapis.com/auth/admin.directory.group.readonly—Scope for only retrieving group, group alias, and member information. This is needed for the Amazon Kendra Identity Crawler.
  - https://www.googleapis.com/auth/admin.directory.user.readonly—Scope for only retrieving users or user aliases. This is needed for listing users in the Amazon Kendra Identity Crawler and for setting ACLs.

  - https://www.googleapis.com/auth/cloud-platform—Scope for generating access token for fetching content of large Google Drive files.
  - https://www.googleapis.com/auth/forms.body.readonly—Scope for fetching data from Google Forms.

  To support the Forms API, add the following additonal scope:

  - https://www.googleapis.com/auth/forms.body.readonly
- Checked each document is unique in Google Drive and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

  > **Note**
  >
  > If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Google Drive authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

  > **Note**
  >
  > We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Google Drive data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Google Drive data source, you must provide the necessary details of your Google Drive data source so that Amazon Kendra can access your data. If you have not yet configured Google Drive for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Google Drive

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Google Drive connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Google Drive connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.
   b. (Optional) **Description**—Enter an optional description for your data source.
   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.
   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.
   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

b. For **Authentication**—Choose between **Google service account** and **OAuth 2.0 authentication** based on your use case.

c. **AWS Secrets Manager secret**—Choose an existing secret, or create a new Secrets Manager secret to store your Google Drive authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

   i. If you chose **Google service account**, enter a name for your secret, the email ID of the admin user or "Service Account User" in your service account configuration (admin email), the email ID of the service account (client email), and the private key that you created in your service account.

      Save and add your secret

   ii. If you chose **OAuth 2.0 authentication**, enter a name for your secret, client ID, client secret, and refresh token that you created in your OAuth account. The user mail id (user whose connection details are configured) will be set as ACL. The connector doesn't set other user/group principal info as ACL due to API limitations.

      Save and add your secret.

d. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

e. (For Google service account authentication users only)

   Identity crawler—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

f. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

g. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. **Sync contents**—Select which options or the content that you want to crawl. You can choose to crawl My Drive (personal folders), Shared Drive (folders shared with you), or both. You can also include file comments.

   b. In **Additional configuration - optional** You can also enter the following optional information:

      i. **Maximum file size**—Set the maximum size limit in MBs of files to crawl.

      ii. **User email**—Add user emails that you want to include or exclude.

      iii. **Shared drives**—Add the shared drive names that you want to include or exclude.

      iv. **Mime types**—Add MIME types that you want to include or exclude.

      v. **Entity regex patterns**—Add regular expression patterns to include or exclude certain attachments for all supported entities. You can add up to 100 patterns.

         You can configure include/exclude regex patterns for **File name**, **File type**, and **File path**.

         - **File name** – The name of the file to include or exclude. For example, to index a file with name `teamroster.txt`, provide `teamroster`.
         - **File type** – The type of the file to include or exclude. For example, .pdf .txt .docx.
         - **File path** – The path of the file to include or exclude. For example, to index files only inside the folder `Products list` of a drive, provide `/Products list`.

   c. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

> **Important**
>
> Google Drive API does not support retrieving comments from a permanently deleted file. Comments from trashed files are retrievable. When a file is trashed, the connector will delete comments from the Amazon Kendra index.

d. In **Sync run schedule**, for **Frequency**—choose how often to sync your data source content and update your index.

e. In **Sync run history**, choose to store auto-generated reports in an Amazon S3 when syncing your data source. This is useful for tracking issues when sycning your data source.

f. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

a. For **Files**—Select from the Amazon Kendra generated default data source fields that you want to map to your index.

> **Note**
>
> Google Drive API does not support creating custom fields. Custom field mapping is not available for the Google Drive connector.

b. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this

page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Google Drive

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `GOOGLEDRIVEV2` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Authentication type**—Specify whether to use service account authentication or OAuth 2.0 authentication.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

> **Important**
>
> Google Drive API does not support retrieving comments from a permanently deleted file. Comments from trashed files are retrievable. When a file is trashed, the connector will delete comments from the Amazon Kendra index.

- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of a Secrets Manager secret that contains the authentication credentials you created in your

Google Drive account. If you use Google service account authentication, the secret is stored in a JSON structure with the following keys:

```
{
    "clientEmail": "user account email",
    "adminAccountEmail": "service account email",
    "privateKey": "private key"
}
```

If you use OAuth 2.0 authentication, the secret is stored in a JSON structure with the following keys:

```
{
    "clientID": "OAuth client ID",
    "clientSecret": "client secret",
    "refreshToken": "refresh token"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Google Drive connector and Amazon Kendra. For more information, see IAM roles for Google Drive data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **My Drives, Shared Drives, Comments**—You can specify whether to crawl these types of content.
- **Inclusion and exclusion filters**—You can specify whether to include or exclude certain user accounts, shared drives, and MIME types.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the

inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Access control list (ACL)**—Specify whether to crawl ACL information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **Field mappings**—Choose to map your Google Drive data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Google Drive template schema.

Notes

- Custom field mapping is not available for Google Drive connector as the Google Drive UI does not support creating custom fields.
- Google Drive API does not support retrieving comments from a permamently deleted file. Comments are retrievable, however, for trashed files. When a file is trashed, the Amazon Kendra connector will delete comments from the Amazon Kendra index.

- Google Drive API does not return comments present in a .docx file.
- If permission for a particular Google document (document, spreadsheet, slide, etc) is set to **General access: Anyone with the link** or **Shared to your specific company domain**, the document will not be visible to Amazon Kendra search users until the user making the query has accessed the document.

IBM DB2

> **Note**
>
> IBM DB2 connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

IBM DB2 is a relational database management system developed by IBM. If you are a IBM DB2 user, you can use Amazon Kendra to index your IBM DB2 data source. The Amazon Kendra IBM DB2 data source connector supports DB2 11.5.7.

You can connect Amazon Kendra to your IBM DB2 data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra IBM DB2 data source connector, see Troubleshooting data sources.

Topics

Supported features

- Field mappings

- User context filtering
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your IBM DB2 data source, make these changes in your IBM DB2 and AWS accounts.

In IBM DB2, make sure you have:

- Noted your database user name and password.

> **Important**
>
> As a best practice, provide Amazon Kendra with read-only database credentials.

- Copied your database host url, port, and instance.
- Checked each document is unique in IBM DB2 and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your IBM DB2 authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your IBM DB2 data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your IBM DB2 data source you must provide details of your IBM DB2 credentials so that Amazon Kendra can access your data. If you have not yet configured IBM DB2 for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to IBM DB2

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **IBM DB2 connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **IBM DB2 connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

a. In **Name and description**, for **Data source name—**Enter a name for your data source. You can include hyphens but not spaces.

b. (Optional) **Description—**Enter an optional description for your data source.

c. In **Default language—**Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag—**Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. In **Source**, enter the following information:

   b. **Host—** Enter the database host name.

   c. **Port—** Enter the database port.

   d. **Instance—** Enter the database instance.

   e. **Enable SSL certificate location—**Choose to enter the Amazon S3 path to your SSL certificate file.

   f. In **Authentication—**enter the following information:

      - **AWS Secrets Manager secret—**Choose an existing secret or create a new Secrets Manager secret to store your IBM DB2 authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

        A. Enter following information in the **Create an AWS Secrets Manager secret window**:

           I. **Secret name—**A name for your secret. The prefix ‘AmazonKendra-IBM DB2-’ is automatically added to your secret name.

           II. For **Database user name**, and **Password—**Enter the authentication credential values you copied from your database.

        B. Choose **Save**.

   g. **Virtual Private Cloud (VPC)—**You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. In **Sync scope**, choose from the following options :

- **SQL query**—Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
- **Primary key column**—Provide the primary key for the database table. This identifies a table within your database.
- **Title column**—Provide the name of the document title column within your database table.
- **Body column**—Provide the name of the document body column within your database table.

b. In **Additional configuration – *optional***, choose from the following options to sync specific content instead of syncing all files:

- **Change-detecting columns**—Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns.
- **User IDs column**—Enter the name of the column which contains User IDs to be allowed access to content.
- **Groups column**—Enter the name of the column that contains groups to be allowed access to content.
- **Source URLs column**—Enter the name of the column which contains Source URLs to be indexed.

  - **Time stamps column—**Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
  - **Time zones column—**Enter the name of the column which contains time zones for the content to be crawled.
  - **Time stamps format—**Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

   c. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

  - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
  - New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

   d. In **Sync run schedule**, for **Frequency—**How often Amazon Kendra will sync with your data source.

   e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. Select from the generated default data source fields—**Document IDs**, **Document titles**, and **Source URLs—**you want to map to Amazon Kendra index.

   b. **Add field—**To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this

page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to IBM DB2

You must specify the following using the TemplateConfiguration API:

- **Data source**—Specify the data source type as `JDBC` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Database type**—You must specify the database type as `db2`.
- **SQL query**—Specify SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials you created in your IBM DB2 account. The secret is stored in a JSON structure with the following keys:

```
{
    "user name": "database user name",
    "password": "password"
```

```
}
```

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the IBM DB2 connector and Amazon Kendra. For more information, see IAM roles for IBM DB2 data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include specific content using user IDs, groups, source URLs, time stamps, and time zones.
- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your IBM DB2 data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see IBM DB2 template schema.

Notes

- Deleted database rows will not be tracked in when Amazon Kendra checks for updated content.
- The size of field names and values in a row of your database can't exceed 400KB.
- If you have a large amount of data in your database data source, and do not want Amazon Kendra to index all your database content after the first sync, you can choose to sync only new, modified, or deleted documents.
- As a best practice, provide Amazon Kendra with read-only database credentials.
- As a best practice, avoid adding tables with sensitive data or personal identifiable information (PII).

Jira

Jira is a project management tool for software development, product management, and bug tracking. You can use Amazon Kendra to index your Jira projects, issues, comments, attachments, worklogs, and statuses.

Amazon Kendra currently only supports Jira Cloud.

You can connect Amazon Kendra to your Jira data source using either the Amazon Kendra console or the JiraConfiguration API. For a list of features supported by each, see Supported features.

For troubleshooting your Amazon Kendra Jira data source connector, see Troubleshooting data sources.

Topics

Supported features

Amazon Kendra Jira data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Jira data source, make these changes in your Jira and AWS accounts.

In Jira, make sure you have:

- Configured API token authentication credentials, which include a Jira ID (user name or email) and a Jira credential (Jira API token). See Atlassian documentation on managing API tokens.

  > **Note**
  >
  > We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Noted the Jira account URL from your Jira account settings. For example, *https://company.atlassian.net/*.
- Checked each document is unique in Jira and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Jira authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Jira data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Jira data source, you must provide the necessary details of your Jira data source so that Amazon Kendra can access your data. If you have not yet configured Jira for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to Jira

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Jira connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Jira connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. **Jira account URL**—Enter your Jira Account URL. For example: *`https://company.atlassian.net/`*.

   b. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

   c. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Jira authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

      i. Enter following information in the **Create an AWS Secrets Manager secret window**:

A. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Jira-' is automatically added to your secret name.

B. For **Jira ID**—Enter the Jira user name or email.

C. For **Password/Token**—Enter the Jira API token configured in Jira.

ii. Save and add your secret.

d. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

e. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

f. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

g. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. **Select which Jira projects to index**—Choose to crawl all project or specific projects.

b. **Additional configuration**—Specify certain statuses, and issue types. Choose to crawl comments, attachments, and worklogs. Use regular expression patterns to include or exclude certain content.

c. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your

data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. **Default data source fields**—Select from the Amazon Kendra generated default data source fields you want to map to your index.

   b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Jira

You must specify the following using the JiraConfiguration API:

- **Data source URL**—Specify your Jira account URL. For example, *company.atlassian.net*.

- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Jira account. The secret is stored in a JSON structure with the following keys:

```
{
    "jiraId": "Jira user name or email",
    "jiraCredential": "Jira API token"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Jira connector and Amazon Kendra. For more information, see IAM roles for Jira data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` as part of the data source configuration. See Configuring Amazon Kendra to use a VPC.
- **Change log**—Whether Amazon Kendra should use the Jira data source change log mechanism to determine if a document must be updated in the index.

> **Note**
>
> Use the change log if you don't want Amazon Kendra to scan all of the documents. If your change log is large, it might take Amazon Kendra less time to scan the documents in the Jira data source than to process the change log. If you are syncing your Jira data source with your index for the first time, all documents are scanned.

- **Inclusion and exclusion filters**—You can specify whether to include or exclude certain files.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Comment, attachments, and work logs**—You can specify whether to crawl certain comments, attachments, and work logs of issues.
- **Projects, Issues, Statuses**—You can specify whether to crawl certain project IDs, issue types, and statuses.
- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Jira data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

Learn more

To learn more about integrating Amazon Kendra with your Jira data source, see:

- Intelligently search your Jira projects with Amazon Kendra Jira Cloud connector

Microsoft Exchange

Microsoft Exchange is an enterprise collaboration tool for messaging, meetings and file sharing. If you are a Microsoft Exchange user, you can use Amazon Kendra to index your Microsoft Exchange data source.

You can connect Amazon Kendra to your Microsoft Exchange data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Microsoft Exchange data source connector, see Troubleshooting data sources.

Supported features

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Microsoft Exchange data source, make these changes in your Microsoft Exchange and AWS accounts.

In Microsoft Exchange, make sure you have:

- Created a Microsoft Exchange account in Office 365.
- Noted your Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal or in your OAuth application.
- Configured an OAuth application in the Azure portal and noted the client ID and client secret or client credentials. See Microsoft tutorial and Registered app example for more information.

> **Note**
>
> When you create or register an app in the Azure portal, the secret ID represents the actual secret value. You must take note or save the actual secret value immediately when creating the secret and app. You can access your secret by selecting the name of your application in the Azure portal and then navigating to the menu option on certificates and secrets.
> You can access your client ID by selecting the name of your application in the Azure portal and then navigating to the overview page. The Application (client) ID is the client ID.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you

re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Added the following permissions for the connector application:

Microsoft Graph	Office 365 Exchange Online
• Mail.Read (Application) • Mail.ReadBasic (Application) • Mail.ReadBasic.All (Application) • Calendars.Read (Application) • User.Read.All (Application) • Contacts.Read (Application) • Notes.Read.All (Application) • Directory.Read.All (Application) • EWS.AccessAsUser.All (Delegated)	full_access_as_app (Application)

- Checked each document is unique in Microsoft Exchange and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Microsoft Exchange authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Microsoft Exchange data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Microsoft Exchange data source, you must provide the necessary details of your Microsoft Exchange data source so that Amazon Kendra can access your data. If you have not yet configured Microsoft Exchange for Amazon Kendra, see [Prerequisites](#).

Console

To connect Amazon Kendra to Microsoft Exchange

1. Sign in to the AWS Management Console and open the [Amazon Kendra console](#).
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Microsoft Exchange connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Microsoft Exchange connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

a. In **Name and description**, for **Data source name—**Enter a name for your data source. You can include hyphens but not spaces.

b. (Optional) **Description—**Enter an optional description for your data source.

c. In **Default language—**Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag—**Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. **Tenant ID—**Enter your Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal or in your OAuth application.

   b. **Authorization—**Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

   c. **AWS Secrets Manager secret—**Choose an existing secret or create a new Secrets Manager secret to store your Microsoft Exchange authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

      i. Enter following information in the **Create an AWS Secrets Manager secret window**:

         A. **Secret name—**A name for your secret. The prefix 'AmazonKendra-Microsoft Exchange

         B. For **Client ID**, **Client secret—**Enter the authentication credentials configured in Microsoft Exchange in the Azure portal.

      ii. Save and add your secret.

   d. **Virtual Private Cloud (VPC)—**You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

   e. **IAM role—**Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

f. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. **User IDs**—Provide the user emails if you want to filter content by certain emails.

   b. **Additional configuration**—Specify the types of content you want to crawl.

      - **Entity types**—You can choose to crawl calendar, OneNotes, or contacts content.
      - **Calendar crawling**—Enter the start and end date to crawl content between certain dates.
      - **Include email**—Enter "to", "from", and email subject lines to filter certain emails you want to crawl.
      - **Shared folders access**—Choose to enable crawling of access control list for access control of your Microsoft Exchange data source.
      - **Regex for domains**—Add regular expression patterns to include or exclude certain email domains.
      - **Regex patterns**—Add regular expression patterns to include or exclude certain files.

   c. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

      - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
      - New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
      - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data

source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. **Default data source fields**—Select from the Amazon Kendra generated default data source fields you want to map to your index.

   > **Note**
   >
   > The Amazon Kendra Microsoft Exchange data source connector doesn't support custom field mappings.

   b. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Microsoft Exchange

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `MSEXCHANGE` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Tenant ID**—You can find your tenant ID in the Properties of your Azure Active Directory Portal or in your OAuth application.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial

sync failed, even if you don't choose full sync as your sync mode option. You can choose between:

  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Microsoft Exchange account. The secret is stored in a JSON structure with the following keys:

```
{
    "clientId": "client ID",
    "clientSecret": "client secret"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Microsoft Exchange connector and Amazon Kendra. For more information, see IAM roles for Microsoft Exchange data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain content.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the

inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Access control list (ACL)**—Specify whether to crawl ACL information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Microsoft Exchange data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Microsoft Exchange template schema.

Learn more

To learn more about integrating Amazon Kendra with your Microsoft Exchange data source, see:

- Index your Microsoft Exchange content using the Exchange connector for Amazon Kendra

Notes

- When Access Control Lists (ACLs) are enabled, the "Sync only new or modified content" option is not available due to Microsoft Exchange API limitations. We recommend using "Full sync" or "New, modified, or deleted content sync" modes instead, or disable ACLs if you need to use this sync mode.

Microsoft OneDrive

Microsoft OneDrive is cloud-based storage service that you can use to store, share, and host your content. You can use Amazon Kendra to index your OneDrive data source.

You can connect Amazon Kendra to your OneDrive data source using the Amazon Kendra console and the OneDriveConfiguration API.

Amazon Kendra has two versions of the OneDrive connector. Supported features of each version include:

Microsoft OneDrive connector V1.0 / OneDriveConfiguration API

- Field mappings
- Inclusion/exclusion filters

Microsoft OneDrive connector V2.0 / TemplateConfiguration API

- User context filtering
- User identity crawler
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

> **Note**
>
> Support for OneDrive connector V1.0 / OneDriveConfiguration API is scheduled to end by June 2023. We recommend using OneDrive connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra OneDrive data source connector, see Troubleshooting data sources.

Topics

- Microsoft OneDrive connector V1.0
- Microsoft OneDrive connector V2.0
- Learn more
- Notes

Microsoft OneDrive connector V1.0

Microsoft OneDrive is a cloud-based storage service that you can use to store, share, and host your content. You can use Amazon Kendra to index your Microsoft OneDrive data source.

> **Note**
>
> Support for OneDrive connector V1.0 / Microsft OneDrive API is scheduled to end by June 2023. We recommend using OneDrive connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra OneDrive data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions

Supported features

- Field mappings
- Inclusion/exclusion filters

Prerequisites

Before you can use Amazon Kendra to index your OneDrive data source, make these changes in your OneDrive and AWS accounts.

In your Azure Active Directory (AD), make sure you have:

- Created an Azure Active Directory (AD) application.
- Used the AD application ID to register a secret key for the application on the AD site. The secret key must contain the application ID and a secret key.
- Copied the AD domain of the organization.
- Added the following application permissions to your AD application on the Microsoft Graph option:

  - Read files in all site collections (File.Read.All)
  - Read all users' full profile (User.Read.All)
  - Read directory data (Directory.Read.All)
  - Read all groups (Group.Read.All)
  - Read items in all site collections (Site.Read.All)
- Copied the list of users whose documents must be indexed. You can choose to provide a list of user names, or you can provide the user names in a file stored in an Amazon S3. After you create the data source, you can:
  - Modify the list of users.
  - Change from a list of users to a list stored in an Amazon S3 bucket.
  - Change the Amazon S3 bucket location of a list of users. If you change the bucket location, you must also update the IAM role for the data source so that it has access to the bucket.

  > **Note**
  >
  > If you store the list of user names in an Amazon S3 bucket, the IAM policy for the data source must provide access to the bucket and access to the key that the bucket was encrypted with, if any.

- Checked each document is unique in OneDrive and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your OneDrive authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your OneDrive data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your OneDrive data source you must provide details of your OneDrive credentials so that Amazon Kendra can access your data. If you have not yet configured OneDrive for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to OneDrive

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **OneDrive connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **OneDrive connector** with the "V2.0" tag.

5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. **OneDrive tenant ID**—Enter the OneDrive tenant ID without the protocol.

   b. **Type of authentication**—Choose between **New** and **Existing**.

   c. i. If you choose **Existing**, select an existing secret for **Select secret**.

      ii. If you choose **New**, enter following information in the **New AWS Secrets Manager secret** section:

         A. **Secret name**—A name for your secret. The prefix 'AmazonKendra-OneDrive-' is automatically added to your secret name.

         B. For **Application ID** and **Application password**—Enter the authentication credential values from your OneDrive account and then choose **Save authentication**.

   d. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

      Note

      IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

   e. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. Choose between **List file** and **Names list** based on your use case.

i. If you choose **List file**, enter the following information:

- **Select location—**Enter the path to your Amazon S3 bucket.

  Add user list file to Amazon S3—Select to add your user list files to your Amazon S3 bucket.

  User local group mappings—Select to use local group mapping to filter your content.

ii. If you choose **Names list**, enter the following information:

- **User name—**Enter up to 10 user drives to index. To add more than 10 users, create a file that contains the names.

  Add another—Choose to add more users.

  User local group mappings—Select to use local group mapping to filter your content.

b. For **Additional configurations—**Add regular expression patterns to include or exclude certain files. You can add up to 100 patterns.

c. In **Sync run schedule**, for **Frequency—**Choose how often Amazon Kendra will sync with your data source.

d. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. For **Default data source fields** and **Additional suggested field mappings—**Select from the Amazon Kendra generated default data source fields you want to map to your index.

   b. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to OneDrive

You must specify the following using the OneDriveConfiguration API:

- **Tenant ID**—Specify the Azure Active Directory domain of the organization.
- **OneDrive Users**—Specify the list of user accounts whose documents should be indexed.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your OneDrive account. The secret is stored in a JSON structure with the following keys:

```
{
    "username": "OAuth client ID",
    "password": "client secret"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the OneDrive connector and Amazon Kendra. For more information, see IAM roles for OneDrive data sources.

You can also add the following optional features:

- **Inclusion and exclusion filters**—Specify whether to include or exclude certain documents.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Field mappings**—Choose to map your OneDrive data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

Microsoft OneDrive connector V2.0

Microsoft OneDrive is cloud-based storage service that you can use to store, share, and host your content. You can use Amazon Kendra to index your OneDrive data source.

You can connect Amazon Kendra to your OneDrive data source using the Amazon Kendra console and the OneDriveConfiguration API.

> **Note**
>
> Support for OneDrive Connector V1.0 / OneDriveConfiguration API is scheduled to end by June 2023. We recommend using OneDrive Connector V2.0 / TemplateConfiguration API. Version 2.0 provides additional ACLs and identity crawler functionality.

For troubleshooting your Amazon Kendra OneDrive data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions

Supported features

Amazon Kendra OneDrive data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your OneDrive data source, make these changes in your OneDrive and AWS accounts.

In OneDrive, make sure you have:

- Created a OneDrive account in Office 365.
- Noted your Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal or in your OAuth application.
- Created an OAuth application in the Azure portal and noted the client ID and client secret or client credentials used for authentication with an AWS Secrets Manager secret. See Microsoft tutorial and Registered app example for more information.

> **Note**
>
> When you create or register an app in the Azure portal, the secret ID represents the actual secret value. You must take note or save the actual secret value immediately when creating the secret and app. You can access your secret by selecting the name of your application in the Azure portal and then navigating to the menu option on certificates and secrets.
> You can access your client ID by selecting the name of your application in the Azure portal and then navigating to the overview page. The Application (client) ID is the client ID.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Used the AD application ID to register a secret key for the application on the AD site. The secret key must contain the application ID and a secret key.
- Copied the AD domain of the organization.
- Added the following permissions to your AD application on the Microsoft Graph option:
  - Read files in all site collections (File.Read.All)
  - Read all users' full profiles(User.Read.All)
  - Read all groups (Group.Read.All)
  - Read all notes (Notes.Read.All)
- Copied the list of users whose documents must be indexed. You can choose to provide a list of user names, or you can provide the user names in a file stored in an Amazon S3. After you create the data source, you can:
  - Modify the list of users.
  - Change from a list of users to a list stored in an Amazon S3 bucket.
  - Change the Amazon S3 bucket location of a list of users. If you change the bucket location, you must also update the IAM role for the data source so that it has access to the bucket.

  > **Note**
  >
  > If you store the list of user names in an Amazon S3 bucket, the IAM policy for the data source must provide access to the bucket and access to the key that the bucket was encrypted with, if any.
  > The OneDrive connector uses **Email from Contact Information** present in the **Onedrive User Properties**. Make sure the user whose data you want to crawl has the email field configured in the **Contact Information** page as for new users this might be blank.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index id.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.
- Stored your OneDrive authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your OneDrive data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index id.

Connection instructions

To connect Amazon Kendra to your OneDrive data source you must provide details of your OneDrive credentials so that Amazon Kendra can access your data. If you have not yet configured OneDrive for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to OneDrive

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **OneDrive connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **OneDrive connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

a. In **Name and description**, for **Data source name—**Enter a name for your data source. You can include hyphens but not spaces.

b. (Optional) **Description—**Enter an optional description for your data source.

c. In **Default language—**Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag—**Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

a. **OneDrive tenant ID—**Enter the OneDrive tenant ID without the protocol.

b. **Authorization—**Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

c. In **Authentication—**Choose between **New** and **Existing**.

d. i. If you choose **Existing**, select an existing secret for **Select secret**.

ii. If you choose **New**, enter following information in the **New AWS Secrets Manager secret** section:

A. **Secret name—**A name for your secret. The prefix ‘AmazonKendra-OneDrive-’ is automatically added to your secret name.

B. For **Client ID** and **Client Secret—**Enter the client ID and client secret.

e. **Virtual Private Cloud (VPC)—**You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

f. **Identity crawler—**Specify whether to turn on Amazon Kendra’s identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra’s identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

g. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

h. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

8. a. For **Sync scope**—Choose which users' OneDrive data to index. You can add a maximum of 10 users manually.

   b. For **Additional configurations**—Add regular expression patterns to include or exclude certain content. You can add up to 100 patterns.

   c. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

   - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
   - New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
   - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

   d. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

   e. Choose **Next**.

9. On the **Set field mappings** page, enter the following information:

   a. **Default data source fields**—Select from the Amazon Kendra generated default data source fields that you want to map to your index.

   b. Choose **Next**.

10. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to OneDrive

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `ONEDRIVEV2` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Tenant ID**—Specify the Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal or in your OAuth application.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source’s mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source’s mechanism for tracking content changes and index content that changed since the last sync.

- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of a Secrets Manager secret that contains the authentication credentials you created in your OneDrive account.

  If you use OAuth 2.0 authentication, the secret is stored in a JSON structure with the following keys:

```
{
    "clientId": "client ID",
    "clientSecret": "client secret"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the OneDrive connector and Amazon Kendra. For more information, see IAM roles for OneDrive data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include or exclude certain files, OneNote sections, and OneNote pages.

  > **Note**
  >
  > Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

- **Field mappings**—You can only map built-in or common index fields for the Amazon Kendra OneDrive connector. Custom field mapping is not available for the OneDrive connector because of API limitations. For more information, see Mapping data source fields.

For a list of other important JSON keys to configure, see OneDrive template schema.

Learn more

To learn more about integrating Amazon Kendra with your OneDrive data source, see:

- Announcing the updated Microsoft OneDrive connector (V2) for Amazon Kendra.

Notes

- When Access Control Lists (ACLs) are enabled, the "Sync only new or modified content" option is not available due to OneDrive API limitations. We recommend using "Full sync" or "New, modified, or deleted content sync" modes instead, or disable ACLs if you need to use this sync mode.

Microsoft SharePoint

SharePoint is a collaborative website building service that you can use to customize web content and create pages, sites, document libraries, and lists. You can use Amazon Kendra to index your SharePoint data source.

Amazon Kendra currently supports SharePoint Online and SharePoint Server (versions 2013, 2016, 2019, and Subscription Edition).

You can connect Amazon Kendra to your SharePoint data source using either the Amazon Kendra console, the TemplateConfiguration API, or the SharePointConfiguration API.

Amazon Kendra has two versions of the SharePoint connector. Supported features of each version include:

SharePoint Connector V1.0 / SharePointConfiguration API

- Field mappings
- User access control
- Inclusion/exclusion filters
- Change log
- Virtual private cloud (VPC)

SharePoint Connector V2.0 / TemplateConfiguration API

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

> **Note**
>
> SharePoint connector V1.0 / SharePointConfiguration API ended in 2023. We recommend migrating to or using SharePoint connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra SharePoint data source connector, see Troubleshooting data sources.

Topics

SharePoint connector V1.0

SharePoint is a collaborative website building service that you can use to customize web content and create pages, sites, document libraries, and lists. If you are a SharePoint user, you can use Amazon Kendra to index your SharePoint data source.

> **Note**
>
> SharePoint connector V1.0 / SharePointConfiguration API ended in 2023. We recommend migrating to or using SharePoint connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra SharePoint data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more

Supported features

- Field mappings
- User access control
- Inclusion/exclusion filters
- Change log
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your SharePoint data source, make these changes in your SharePoint and AWS accounts.

You are required to provide authentication credentials, which you securely store in an AWS Secrets Manager secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-

use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

In SharePoint, make sure you have:

- Noted the URL of the SharePoint sites you want to index.
- **For SharePoint Online:**
  - Noted your basic authentication credentials containing a user name and password with site admin permissions.
  - **Optional:** Generated OAuth 2.0 credentials containing a user name, password, client ID, and client secret.
  - Deactivated **Security Defaults** in your Azure portal using an administrative user. For more information on managing security default settings in the Azure portal, see Microsoft documentation on how to enable/disable security defaults.
- **For SharePoint Server:**
  - Noted your SharePoint Server domain name (the NetBIOS name in your Active Directory). You use this, along with your SharePoint basic authentication user name and password, to connect SharePoint Server to Amazon Kendra.

  > **Note**
  >
  > If you use SharePoint Server and need to convert your Access Control List (ACL) to email format for filtering on user context, provide the LDAP server URL and LDAP search base. Or you can use the directory domain override. The LDAP server URL is the full domain name and the port number (for example, ldap://example.com:389). The LDAP search base are the domain controllers 'example' and 'com'. With the directory domain override, you can use the email domain instead of using LDAP server URL and LDAP search base. For example, the email domain for username@example.com is 'example.com'. You can use this override if you aren't concerned about validating your domain and simply want to use your email domain.

- Added the following permissions to your SharePoint account:

  For SharePoint lists

  - Open Items—View the source of documents with server-side file handlers.

  - View Application Pages—View forms, views, and application pages. Enumerate lists.
  - View Items—View items in lists and documents in document libraries.
  - View Versions—View past versions of a list item or document.

  For SharePoint websites

  - Browse Directories—Enumerate files and folders in a website using SharePoint Designer and Web DAV interface.
  - Browse User Information—View information about users of the website.
  - Enumerate Permissions—Enumerate permissions on the website, list, folder, document, or list item.
  - Open—Open a website, list, or folder to access items inside the container.
  - Use Client Integration Features—Use SOAP, WebDAV, the client object model, or SharePoint Designer interfaces to access the website.
  - Use Remote Interfaces—Use features that launch client applications.
  - View Pages—View pages on a website.
- Checked each document is unique in SharePoint and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- [Created an Amazon Kendra index](#) and, if using the API, noted the index ID.
- [Created an IAM role](#) for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your SharePoint authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your SharePoint data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your SharePoint data source you must provide details of your SharePoint credentials so that Amazon Kendra can access your data. If you have not yet configured SharePoint for Amazon Kendra see [Prerequisites].

Console

To connect Amazon Kendra to SharePoint

1. Sign in to the AWS Management Console and open the [Amazon Kendra console].
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **SharePoint connector v1.0**, and then choose **Add data source**.
5. On the **Specify data source details** page, enter the following information:

a. In **Name and description**, for **Data source name—**Enter a name for your data source. You can include hyphens but not spaces.

b. (Optional) **Description—**Enter an optional description for your data source.

c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag—**Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. For **Hosting method**—Choose between **SharePoint Online** and **SharePoint Server**.

      i. For **SharePoint Online—**Enter the **Site URLs specific to your SharePoint repository**.

      ii. For **SharePoint Server—**Choose your **SharePoint version**, enter **Site URLs specific to your SharePoint repository**, and enter the Amazon S3 path to your **SSL certificate location**.

   b. (SharePoint Server only) For **Web proxy—**Enter the **Host name** and **Port number** of your internal SharePoint instance. The port number should be a numeric value between 0 and 65535.

   c. For **Authentication—**Choose between the following options based on your use case:

      i. For SharePoint Online—Choose between **Basic authentication** and **OAuth 2.0 authentication**.

      ii. For SharePoint Server—Choose between **None**, **LDAP**, and **Manual**.

   d. For **AWS Secrets Manager secret—**Choose an existing secret or create a new Secrets Manager secret to store your SharePoint authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens. You must enter a **Secret name**. The prefix 'AmazonKendra-SharePoint-' is automatically added to your secret name.

   e. Enter following other information in the **Create an AWS Secrets Manager secret window**:

i. Choose from the following SharePoint Cloud authentication options, based on your use case:

   A. **Basic authentication—**Enter your SharePoint account user name as **User name** and SharePoint account password as **Password**.

   B. **OAuth 2.0 authentication—**Enter your SharePoint account user name as **User name**, SharePoint account password as **Password**, your auto-generated unique SharePoint ID as **Client ID**, and the shared secret string used by both SharePoint and Amazon Kendra as **Client secret**.

ii. Choose from the following SharePoint Server authentication options, based on your use case:

   A. **None—**Enter your SharePoint account user name as **User name**, your SharePoint account password as **Password**, and your **Server Domain Name**.

   B. **LDAP—**Enter your SharePoint account user name as **User name**, SharePoint account password as **Password**, your **LDAP Server Endpoint** (including protocol and port number, for example `ldap://example.com:389`), and your **LDAP Search Base** (for example, `dc=example, dc=com`).

   C. **Manual—**Enter your SharePoint account user name as **User name**, your SharePoint account password as **Password**, and your **Email Domain Override** (email domain of directory user or group).

iii. Choose **Save**.

f. **Virtual Private Cloud (VPC)—** You must also add **Subnets** and **VPC security groups**.

> **Note**
>
> You must use a VPC if you use SharePoint Server. Amazon VPC is optional for other SharePoint versions.

g. **IAM role—**Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

   h. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. **Use Change log**—Select to update your index instead of syncing all your files.

   b. **Crawl attachments**—Select to crawl attachments.

   c. **Use local group mappings**—Select to make sure that documents are properly filtered.

   d. **Additional configuration**—Add regular expression patterns to include or exclude certain files. You can add up to 100 patterns.

   e. In **Sync run schedule** for **Frequency**—How often Amazon Kendra will sync with your data source.

   f. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. **Amazon Kendra default field mappings**—Select from the Amazon Kendra generated default data source fields you want to map to your index.

   b. For **Custom field mappings**—Add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to SharePoint

You must specify the following using [SharePointConfiguration](#) API:

- **SharePoint Version**—Specify the SharePoint version you use when configuring SharePoint. This is the case no matter if you use SharePoint Server 2013, SharePoint Server 2016, SharePoint Server 2019, or SharePoint Online.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of a Secrets Manager secret that contains the authentication credentials you created in your SharePoint account.The secret is stored in a JSON structure.

  For **SharePoint Online basic authentication**, the following is the minimum JSON structure that must be in your secret:

  ```
  {
      "userName": "user name",
      "password": "password"
  }
  ```

  For **SharePoint Online OAuth 2.0 authentication**, the following is the minimum JSON structure that must be in your secret:

  ```
  {
      "userName": "SharePoint account user name"",
      "password": "SharePoint account password",
      "clientId": "SharePoint auto-generated unique client id",
      "clientSecret": "secret string shared by Amazon Kendra and SharePoint to
   authorize communications"
  }
  ```

  For **SharePoint Server basic authentication**, the following is the minimum JSON structure that must be in your secret:

  ```
  {
      "userName": "user name",
      "password": "password",
      "domain": "server domain name"
  }
  ```

  For **SharePoint Server LDAP authentication** (if you need to convert your access control list (ACL) to email format for filtering on user context you can include the LDAP server URL and LDAP search base in your secret), the following is the minimum JSON structure that must be in your secret:

```
{
    "userName": "user name",
    "password": "password",
    "domain": "server domain name"
    "ldapServerUrl": "ldap://example.com:389",
    "ldapSearchBase": "dc=example,dc=com"
}
```

For **SharePoint Server Manual authentication**, the following is the minimum JSON structure that must be in your secret::

```
{
    "userName": "user name",
    "password": "password",
    "domain": "server domain name",
    "emailDomainOverride": "example.com"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the SharePoint connector and Amazon Kendra. For more information, see IAM roles for SharePoint data sources.
- **Amazon VPC**—If you use SharePoint Server, specify `VpcConfiguration` as part of the data source configuration. See Configuring Amazon Kendra to use a VPC.

You can also add the following optional features:

- **Web proxy**—Whether to connect to your SharePoint site URLs via a web proxy. You can use this option only for SharePoint Server.
- **Indexing lists**—Whether Amazon Kendra should index the contents of attachments to SharePoint list items.
- **Change log**—Whether Amazon Kendra should use the SharePoint data source change log mechanism to determine if a document must be updated in the index.

> **Note**
>
> Use the change log if you don't want Amazon Kendra to scan all of the documents. If your change log is large, it might take Amazon Kendra less time to scan the

documents in the SharePoint data source than to process the change log. If you are syncing your SharePoint data source with your index for the first time, all documents are scanned.

- **Inclusion and exclusion filters**—You can specify whether to include or exclude certain content.

  Note

  Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Field mappings**—Choose to map your SharePoint data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

  Note

  The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

Learn more

To learn more about integrating Amazon Kendra with your SharePoint data source, see:

- Getting started with the Amazon Kendra SharePoint Online connector

SharePoint connector V2.0

SharePoint is a collaborative website building service that you can use to customize web content and create pages, sites, document libraries, and lists. You can use Amazon Kendra to index your SharePoint data source.

Amazon Kendra currently supports SharePoint Online and SharePoint Server (2013, 2016, 2019, and Subscription Edition).

> **Note**
>
> SharePoint connector V1.0 / SharePointConfiguration API ended in 2023. We recommend migrating to or using SharePoint connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra SharePoint data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Notes

Supported features

Amazon Kendra SharePoint data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your SharePoint data source, make these changes in your SharePoint and AWS accounts.

You are required to provide authentication credentials, which you securely store in an AWS Secrets Manager secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

In SharePoint Online, make sure you have:

- Copied your SharePoint instance URLs. The format for the host URL you enter is *`https://yourdomain.com/sites/mysite`*. Your URL must start with `https`.
- Copied the domain name of your SharePoint instance URL.
- Noted your basic authentication credentials containing the user name and password with site admin permissions to connect to SharePoint Online.
- Deactivated **Security Defaults** in your Azure portal using an administrative user. For more information on managing security default settings in the Azure portal, see Microsoft documentation on how to enable/disable security defaults.
- Deactivated multi-factor authentication (MFA) in your SharePoint account, so that Amazon Kendra is not blocked from crawling your SharePoint content.
- **If using authentication type other than Basic authentication:** Copied the tenant ID of your SharePoint instance. For details on how to find your tenant ID, see Find your Microsoft 365 tenant ID.
- If you need to migrate to cloud user authentication with Microsoft Entra, see Microsoft documentation on cloud authentication.
- **For OAuth 2.0 authentication and OAuth 2.0 refresh token authentication:** Noted your **Basic authentication** credentials containing the user name and password you use to connect to SharePoint Online and the client ID and client secret generated after registering SharePoint with Azure AD.

- **If you're not using ACL**, added the following permissions:

Microsoft Graph	SharePoint
• Notes.Read.All (Application)—Read all OneNote notebooks • Sites.Read.All (Application)—Read items in all site collections	• AllSites.Read (Delegated)—Read items in all site collections

> **Note**
>
> Note.Read.All and Sites.Read.All are required only if you want to crawl OneNote Documents.
> If you want to crawl specific sites, the permission can be restricted to specific sites rather than all sites available in the domain. You configure **Sites.Selected (Application)** permission. With this API permission, you need to set access permission on every site explicitly through Microsoft Graph API. For more information, see Microsoft's blog on Sites.Selected permissions.

- **If you're using ACL**, added the following permissions:

Microsoft Graph	SharePoint
• Group.Member.Read.All (Application)—Read all group memberships • Notes.Read.All (Application)—Read all OneNote notebooks • Sites.FullControl.All (Delegated)—Required to retrieve ACLs of the documents • Sites.Read.All (Application)—Read items in all site collections • User.Read.All (Application)—Read all users' full profiles	• AllSites.Read (Delegated)—Read items in all site collections

> **Note**
>
> GroupMember.Read.All and User.Read.All are required only if **Identity crawler** is activated.
> If you want to crawl specific sites, the permission can be restricted to specific sites rather than all sites available in the domain. You configure **Sites.Selected (Application)** permission. With this API permission, you need to set access permission on every site explicitly through Microsoft Graph API. For more information, see [Microsoft's blog on Sites.Selected permissions](#).

- **For Azure AD App-Only authentication:** Private key and the Client ID you generated after registering SharePoint with Azure AD. Also note the X.509 certificate.
  - **If you're not using ACL**, added the following permissions:

SharePoint	
• Sites.Read.All (Application)—Required to access items and lists in all site collections	

> **Note**
>
> If you want to crawl specific sites, the permission can be restricted to specific sites rather than all sites available in the domain. You configure **Sites.Selected (Application)** permission. With this API permission, you need to set access permission on every site explicitly through Microsoft Graph API. For more information, see [Microsoft's blog on Sites.Selected permissions](#).

  - **If you're using ACL**, added the following permissions:

SharePoint	
• Sites.FullControl.All (Application)—Requ ired to retrieve ACLs of the documents	

> **Note**
>
> If you want to crawl specific sites, the permission can be restricted to specific sites rather than all sites available in the domain. You configure **Sites.Selected (Application)** permission. With this API permission, you need to set access permission on every site explicitly through Microsoft Graph API. For more information, see [Microsoft's blog on Sites.Selected permissions](#).

- **For SharePoint App-Only authentication:** Noted your SharePoint client ID and client secret generated while granting permission to SharePoint App Only, and your Client ID and Client secret generated when you registered your SharePoint app with Azure AD.

> **Note**
>
> SharePoint App-Only Authentication is *not* supported for SharePoint 2013 version.

- **(Optional) If you're crawling OneNote documents and using Identity crawler**, added the following permissions:

Microsoft Graph	
• GroupMember.Read.All (Application)—Read all group memberships • Notes.Read.All (Application)—Read all OneNote notebooks • Sites.Read.All (Application)—Read items in all site collections • User.Read.All (Application)—Read all users' full profiles	

> **Note**
>
> No API permissions are required for crawling entities using **Basic authentication** and SharePoint **App-only authentication**.

In SharePoint Server, make sure you have:

- Copied your SharePoint instance URLs and the domain name of your SharePoint URLs. The format for the host URL you enter is *https://yourcompany/sites/mysite*. Your URL must start with `https`.

> **Note**
>
> (On-premise/server) Amazon Kendra checks if the endpoint information included in AWS Secrets Manager is the same the endpoint information specified in your data source configuration details. This helps protect against the confused deputy problem, which is a security issue where a user doesn't have permission to perform an action but uses Amazon Kendra as a proxy to access the configured secret and perform the action. If you later change your endpoint information, you must create a new secret to sync this information.

- Deactivated multi-factor authentication (MFA) in your SharePoint account, so that Amazon Kendra is not blocked from crawling your SharePoint content.
- If using **SharePoint App-Only authentication** for access control:
  - Copied the SharePoint client ID generated when you registered App Only at Site Level. Client ID format is ClientId@TenantId. For example, *ffa956f3-8f89-44e7-b0e4-49670756342c@888d0b57-69f1-4fb8-957f-e1f0bedf82fe*.
  - Copied the SharePoint client secret generated when you registered App Only at Site Level.

  Note: Because client IDs and client secrets are generated for single sites only when you register SharePoint Server for App Only authentication, only one site URL is supported for SharePoint App Only authentication.

> **Note**
>
> SharePoint App-Only Authentication is *not* supported for SharePoint 2013 version.

- If using **Email ID with Custom Domain** for access control:
  - Noted your custom email domain value—for example: "*amazon.com*".
- If using **Email ID with Domain from IDP** authorization, copied your:
  - LDAP Server Endpoint (endpoint of LDAP server including protocol and port number). For example: *ldap://example.com:389*.
  - LDAP Search Base (search base of the LDAP user). For example: *CN=Users,DC=sharepoint,DC=com*.
  - LDAP user name and LDAP password.
- Either configured NTLM authentication credentials **or** configured Kerberos authentication credentials containing a user name (SharePoint account user name) and password (SharePoint account password).

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your SharePoint authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your SharePoint data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your SharePoint data source, you must provide details of your SharePoint credentials so that Amazon Kendra can access your data. If you have not yet configured SharePoint for Amazon Kendra see Prerequisites.

Console: SharePoint Online

To connect Amazon Kendra to SharePoint Online

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **SharePoint connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **SharePoint connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.
   b. (Optional) **Description**—Enter an optional description for your data source.
   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.
   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. **Hosting Method**—Choose **SharePoint Online**.

   b. **Site URLs specific to your SharePoint repository**—Enter the SharePoint host URLs. The format for the host URLs you enter is *`https://yourdomain.sharepoint.com/sites/mysite`*. The URL must start with `https` protocol. Separate URLs with a new line. You can add up to 100 URLs.

   c. **Domain**—Enter the SharePoint domain. For example, the domain in the URL *`https://yourdomain.sharepoint.com/sites/mysite`* is *`yourdomain`*.

   d. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

      You can also choose the type of user ID, whether the user principal name or the user email fetched from the Azure Portal. If you don't specify, email is used by default.

   e. **Authentication**—Choose either basic, OAuth 2.0, Azure AD App-Only authentication, SharePoint App-Only authentication, or OAuth 2.0 refresh token authentication. You either choose an existing AWS Secrets Manager secret to store your authentication credentials, or create a secret.

      i. If using **Basic Authentication**, your secret must include a secret name, SharePoint user name and password.

      ii. If using **OAuth 2.0 authentication**, your secret must include the SharePoint tenant ID, secret name, SharePoint user name, password, Azure AD client ID generated when you register SharePoint in Azure AD, and Azure AD client secret generated when you register SharePoint in Azure AD.

      iii. If using **Azure AD App-Only authentication**, your secret must include the SharePoint tenant ID, Azure AD self-signed X.509 certificate, secret name, Azure AD client ID generated when you register SharePoint in Azure AD, and private key to authenticate the connector for Azure AD.

      iv. If using **SharePoint App-Only authentication**, your secret must include the SharePoint tenant ID, secret name, SharePoint client ID you generated when you registered App Only at Tenant Level, SharePoint client secret generated when your register for App Only at Tenant Level, Azure AD client ID generated when you register SharePoint in Azure AD, and Azure AD client secret generated when you register SharePoint to Azure AD.

The SharePoint client ID format is *ClientID@TenantId*. For example, *ffa956f3-8f89-44e7-b0e4-49670756342c@888d0b57-69f1-4fb8-957f-e1f0bedf82fe*.

v. If using **OAuth 2.0 refresh token authentication**, your secret must include the SharePoint tenant ID, secret name, unique Azure AD client ID generated when you register SharePoint in Azure AD, Azure AD client secret generated when you register SharePoint to Azure AD, refresh token generated to connect Amazon Kendra to SharePoint.

f. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

g. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

You can also choose to crawl local group mapping or Azure Active Directory group mapping.

> **Note**
>
> AD Group mapping crawling is available only for OAuth 2.0, OAuth 2.0 refresh token, and SharePoint App Only authentication.

h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

   i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. In **Sync scope**, choose from the following options :

      i. **Select entities**—Choose the entities you want to crawl. You can select to crawl **All** entities or any combination of **Files**, **Attachments**, **Links Pages**, **Events**, **Comments**, and **List Data**.

      ii. In **Additional configuration**, for **Entity regex patterns**—Add regular expression patterns for **Links**, **Pages**, and **Events** to include specific entities instead of syncing all your documents.

      iii. **Regex patterns**—Add regular expression patterns to include or exclude files by **File path**, **File name**, **File type**, **OneNote section name**, and **OneNote page name** instead of syncing all your documents. You can add up to 100.

      > **Note**
      >
      > OneNote crawling is available only for OAuth 2.0, OAuth 2.0 refresh token, and SharePoint App Only authentication.

   b. For **Sync mode** choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is synced by default.

      - **Full sync**—Sync all content regardless of the previous sync status.
      - **New or modified documents sync**—Sync only new or modified documents.
      - **New, modified, or deleted documents sync**—Sync only new, modified, and deleted documents.

   c. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

   d. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. **Default data source fields**—Select from the Amazon Kendra generated default data source fields that you want to map to your index.

   b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

Console: SharePoint Server

To connect Amazon Kendra to SharePoint

1. Sign in to the AWS Management Console and open the [Amazon Kendra console](https://console.aws.amazon.com/kendra/).

2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.

4. On the **Add data source** page, choose **SharePoint connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **SharePoint connector** with the "V2.0" tag.

5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. **Hosting Method**—Choose **SharePoint Server**.

   b. **Choose SharePoint Version**—Choose either **SharePoint 2013**, **SharePoint 2016**, **SharePoint 2019**, and **SharePoint (Subscription Edition)**.

   c. **Site URLs specific to your SharePoint repository**—Enter the SharePoint host URLs. The format for the host URLs you enter is *`https://yourcompany/sites/mysite`*. The URL must start with `https` protocol. Separate URLs with a new line. You can add up to 100 URLs.

   d. **Domain**—Enter the SharePoint domain. For example, the domain in the URL *`https://yourcompany/sites/mysite`* is *`yourcompany`*

   e. **SSL certificate location**—Enter the Amazon S3 path to your SSL certificate file.

   f. (Optional) For **Web proxy**—Enter the host name (without the `http://` or `https://` protocol), and the port number used by the host URL transport protocol. The numeric value of the port number must be between 0 and 65535.

   g. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see [User context filtering](#).

      For SharePoint Server you can choose from the following ACL options:

      i. **Email ID with Domain from IDP**—User ID is based on email IDs with their domains fetched from the underlying identity provider (IDP). You provide the IDP connection details in your Secrets Manager secret as part of **Authentication**.

      ii. **Email ID with Custom Domain**—User ID is based on the custom email domain value. For example, "*`amazon.com`*". The email domain will be used to construct the email ID for access control. You must enter your custom email domain.

iii. **Domain\User with Domain**—User ID is constructed using a Domain \User ID format. You need to provide a valid domain name. For example: *"sharepoint2019"* to construct access control.

h. For **Authentication**, choose either SharePoint App-Only authentication, NTLM authentication, or Kerberos authentication. You either choose an existing AWS Secrets Manager secret to store your authentication credentials, or create a secret.

i. If using **NTLM authentication** or **Kerberos authentication**, you secret must include a secret name, SharePoint user name and password.

If using **Email ID with Domain from IDP**, also enter your:

- **LDAP Server Endpoint**—Endpoint of LDAP server, including protocol and port number. For example: *ldap://example.com:389*.
- **LDAP Search Base**—Search base of LDAP user. For example: *CN=Users,DC=sharepoint,DC=com*.
- **LDAP username**—Your LDAP user name.
- **LDAP Password**—Your LDAP password.

ii. If using **SharePoint App-Only authentication**, your secret must include a secret name, SharePoint client ID you generated when you registered App Only at Site Level, SharePoint client secret generated when your register for App Only at Site Level.

The SharePoint client ID format is *ClientID@TenantId*. For example, *ffa956f3-8f89-44e7-b0e4-49670756342c@888d0b57-69f1-4fb8-957f-e1f0bedf82fe*.

Note: Because client IDs and client secrets are generated for single sites only when you register SharePoint Server for App Only authentication, only one site URL is supported for SharePoint App Only authentication.

If using **Email ID with Domain from IDP**, also enter your:

- **LDAP Server Endpoint**—Endpoint of LDAP server, including protocol and port number. For example: *ldap://example.com:389*.
- **LDAP Search Base**—Search base of LDAP user. For example: *CN=Users,DC=sharepoint,DC=com*.

- **LDAP username**—Your LDAP user name.
- **LDAP Password**—Your LDAP password.

i. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

j. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

   You can also choose to crawl local group mapping or Azure Active Directory group mapping.

   > **Note**
   >
   > AD Group mapping crawling is available only SharePoint App Only authentication.

k. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

   > **Note**
   >
   > IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

l. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. In **Sync scope**, choose from the following options :

i. **Select entities**—Choose the entities you want to crawl. You can select to crawl **All** entities or any combination of **Files**, **Attachments**, **Links Pages**, **Events**, and **List Data**.

ii. In **Additional configuration**, for **Entity regex patterns**—Add regular expression patterns for **Links**, **Pages**, and **Events** to include specific entities instead of syncing all your documents.

iii. **Regex patterns**—Add regular expression patterns to include or exclude files by **File path File name File type**, **OneNote section name**, and **OneNote page name** instead of syncing all your documents. You can add up to 100.

> **Note**
>
> OneNote crawling is available only for SharePoint App Only authentication.

b. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

c. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

d. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

a. **Default data source fields**—Select from the Amazon Kendra generated default data source fields that you want to map to your index.

b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to SharePoint

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `SHAREPOINTV2` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Repository Endpoint Metadata**—Specify the `tenantID` `domain` and `siteUrls` of your SharePoint instance.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

> **Note**
>
> Identity crawler is available only when you set `crawlAcl` to `true`.

- **Repository Additional Properties**—Specify the:
  - (For Azure AD) `s3bucketName` and `s3certificateName` you use to store your Azure AD self-signed X.509 certificate.
  - Authentication type (`auth_Type`) you use, whether `OAuth2`, `OAuth2App`, `OAuth2Certificate`, `Basic`, `OAuth2_RefreshToken`, `NTLM`, and `Kerberos`.
  - Version (`version`) you use, whether `Server` or `Online`. If you use `Server` you can futher specify the `onPremVersion` as `2013`, `2016`, `2019`, or `SubscriptionEdition`.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of a Secrets Manager secret that contains the authentication credentials you created in your SharePoint account.

  If you use SharePoint Online, you can choose between Basic, OAuth 2.0, Azure AD App-only and SharePoint App Only authentication. The following are the minimum JSON structure that must be in your secret for each authentication option:

  - **Basic authentication**

```
{
    "userName": "SharePoint account user name",
    "password": "SharePoint account password"
}
```

  - **OAuth 2.0 authentication**

```
{
    "clientId": "client id generated when registering SharePoint with Azure AD",
    "clientSecret": "client secret generated when registering SharePoint with Azure AD",
    "userName": "SharePoint account user name",
    "password": "SharePoint account password"
}
```

- **Azure AD App-Only authentication**

```
{
    "clientId": "client id generated when registering SharePoint with Azure AD",
    "privateKey": "private key to authorize connection with Azure AD"
}
```

- **SharePoint App-Only authentication**

```
{
    "clientId": "client id generated when registering SharePoint for App Only at Tenant Level",
    "clientSecret": "client secret generated when registering SharePoint for App Only at Tenant Level",
    "adClientId": "client id generated while registering SharePoint with Azure AD",
    "adClientSecret": "client secret generated while registering SharePoint with Azure AD"
}
```

- **OAuth 2.0 refresh token authentication**

```
{
    "clientId": "client id generated when registering SharePoint with Azure AD",
    "clientSecret": "client secret generated when registering SharePoint with Azure AD",
    "refreshToken": "refresh token generated to connect to SharePoint"
}
```

If you use SharePoint Server, you can choose between SharePoint App-Only authentication, NTLM authentication, and Kerberos authentication. The following are the minimum JSON structure that must be in your secret for each authentication option:

- **SharePoint App-Only authentication**

```
{
    "siteUrlsHash": "Hash representation of SharePoint site URLs",
    "clientId": "client id generated when registering SharePoint for App Only at
 Site Level",
    "clientSecret": "client secret generated when registering SharePoint for App
 Only at Site Level"
}
```

- **SharePoint App-Only authentication with domain from IDP authorization**

```
{
    "siteUrlsHash": "Hash representation of SharePoint site URLs",
    "clientId": "client id generated when registering SharePoint for App Only at
 Site Level",
    "clientSecret": "client secret generated when registering SharePoint for App
 Only at Site Level",
    "ldapUrl": "LDAP Account url eg. ldap://example.com:389",
    "baseDn": "LDAP Account base dn eg. CN=Users,DC=sharepoint,DC=com",
    "ldapUser": "LDAP account user name",
    "ldapPassword": "LDAP account password"
}
```

- **(Server only) NTLM or Kerberos authentication**

```
{
    "siteUrlsHash": "Hash representation of SharePoint site URLs",
    "userName": "SharePoint account user name",
    "password": "SharePoint account password"
}
```

- **(Server only) NTLM or Kerberos authentication with domain from IDP authorization**

```
{
    "siteUrlsHash": "Hash representation of SharePoint site URLs",
    "userName": "SharePoint account user name",
    "password": "SharePoint account password",
    "ldapUrl": "ldap://example.com:389",
    "baseDn": "CN=Users,DC=sharepoint,DC=com",
    "ldapUser": "LDAP account user name",
    "ldapPassword": "LDAP account password"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the SharePoint connector and Amazon Kendra. For more information, see IAM roles for SharePoint data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include or exclude certain files, OneNotes, and other content.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Field mappings**—Choose to map your SharePoint data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see SharePoint template schema.

Notes

- The connector supports custom field mappings only for the **Files** entity.

- For all SharePoint Server versions, the ACL token must be in lower case. For **Email with Domain from IDP** and **Email ID with Custom Domain** ACL, for example: *user@sharepoint2019.com*. For **Domain\User with Domain** ACL, for example: *sharepoint2013\user*.
- When Access Control Lists (ACLs) are enabled, the "Sync only new or modified content" option is not available due to SharePoint API limitations. We recommend using "Full sync" or "New, modified, or deleted content sync" modes instead, or disable ACLs if you need to use this sync mode.
- The connector does not support change log mode/**New or modified content sync** for SharePoint 2013.
- If an entity name has a % character in its name, the connector will skip these files due to API limitations.
- OneNote can only be crawled by the connector using a Tenant ID, and with OAuth 2.0, OAuth 2.0 refresh token, or SharePoint App Only authentication activated for SharePoint Online.
- The connector crawls the first section of a OneNote document using its default name only, even if the document is renamed.
- The connector crawls links in SharePoint 2019, SharePoint Online, and Subscription Edition, only if **Pages** and **Files** are selected as entities to be crawled in addition to **Links**.
- The connector crawls links in SharePoint 2013 and SharePoint 2016 if **Links** is selected as an entity to be crawled.
- The connector crawls list attachments and comments only when **List Data** is also selected as an entity to be crawled.
- The connector crawls event attachments only when **Events** is also selected as an entity to be crawled.
- For SharePoint Online version, the ACL token will be in lower case. For example, if **User principal name** is *MaryMajor@domain.com* in Azure portal, the ACL token in the SharePoint Connector will be *marymajor@domain.com*.
- In **Identity Crawler** for SharePoint Online and Server, if you want to crawl nested groups, you have to activate Local as well as AD Group Crawling.
- If you're using SharePoint Online, and the User Principal Name in your Azure Portal is a combination of upper case and lower case, the SharePoint API internally converts it to lower case. Because of this, the Amazon Kendra SharePoint connector sets ACL in lower case.

Microsoft SQL Server

Note

Microsoft SQL Server connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

Microsoft SQL Server is an relational database management system (RDBMS) developed by Microsoft. If you are a Microsoft SQL Server user, you can use Amazon Kendra to index your Microsoft SQL Server data source. The Amazon Kendra Microsoft SQL Server data source connector supports MS SQL Server 2019.

You can connect Amazon Kendra to your Microsoft SQL Server data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Microsoft SQL Server data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Notes

Supported features

- Field mappings
- User context filtering
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Microsoft SQL Server data source, make these changes in your Microsoft SQL Server and AWS accounts.

In Microsoft SQL Server, make sure you have:

- Noted your database user name and password.

> **Important**
>
> As a best practice, provide Amazon Kendra with read-only database credentials.

- Copied your database host url, port, and instance.
- Checked each document is unique in Microsoft SQL Server and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Microsoft SQL Server authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Microsoft SQL Server data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Microsoft SQL Server data source you must provide details of your Microsoft SQL Server credentials so that Amazon Kendra can access your data. If you have not yet configured Microsoft SQL Server for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Microsoft SQL Server

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Microsoft SQL Server connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Microsoft SQL Server connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. In **Source**, enter the following information:

   b. **Host**— Enter the database host name.

   c. **Port**— Enter the database port.

   d. **Instance**— Enter the database instance.

   e. **Enable SSL certificate location**—Choose to enter the Amazon S3 path to your SSL certificate file.

   f. In **Authentication**—enter the following information:

      - **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Microsoft SQL Server authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

        A. Enter following information in the **Create an AWS Secrets Manager secret window**:

           I. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Microsoft SQL Server-' is automatically added to your secret name.

           II. For **Database user name**, and **Password**—Enter the authentication credential values you copied from your database.

        B. Choose **Save**.

   g. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

   h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

      > **Note**
      >
      > IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

   i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. In **Sync scope**, choose from the following options :

- **SQL query**—Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.

> **Note**
>
> If a table name includes special characters (non alphanumeric) in the name, you must use square brackets around the table name. For example, `select * from [my-database-table]`

- **Primary key column**—Provide the primary key for the database table. This identifies a table within your database.
- **Title column**—Provide the name of the document title column within your database table.
- **Body column**—Provide the name of the document body column within your database table.

b. In **Additional configuration – *optional***, choose from the following options to sync specific content instead of syncing all files:

- **Change-detecting columns**—Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns.
- **User IDs column**—Enter the name of the column which contains User IDs to be allowed access to content.
- **Groups column**—Enter the name of the column that contains groups to be allowed access to content.
- **Source URLs column**—Enter the name of the column which contains Source URLs to be indexed.
- **Time stamps column**—Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
- **Time zones column**—Enter the name of the column which contains time zones for the content to be crawled.

- **Time stamps format—**Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

c. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency—**How often Amazon Kendra will sync with your data source.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. Select from the generated default data source fields—**Document IDs**, **Document titles**, and **Source URLs**—you want to map to Amazon Kendra index.

   b. **Add field—**To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Microsoft SQL Server

You must specify the following using the TemplateConfiguration API:

- **Data source**—Specify the data source type as JDBC when you use the TemplateConfiguration JSON schema. Also specify the data source as TEMPLATE when you call the CreateDataSource API.
- **Database type**—You must specify the database type as sqlserver.
- **SQL query**—Specify SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.

> **Note**
>
> If a table name includes special characters (non alphanumeric) in the name, you must use square brackets around the table name. For example, *select * from [my-database-table]*

- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - FORCED_FULL_CRAWL to freshly index all content, replacing existing content each time your data source syncs with your index.
  - FULL_CRAWL to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - CHANGE_LOG to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials you created in your Microsoft SQL Server account. The secret is stored in a JSON structure with the following keys:

```
{
    "user name": "database user name",
```

```
    "password": "password"
}
```

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Microsoft SQL Server connector and Amazon Kendra. For more information, see IAM roles for Microsoft SQL Server data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include specific content using user IDs, groups, source URLs, time stamps, and time zones.
- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Microsoft SQL Server data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Microsoft SQL Server template schema.

Notes

- Deleted database rows will not be tracked in when Amazon Kendra checks for updated content.
- The size of field names and values in a row of your database can't exceed 400KB.
- If you have a large amount of data in your database data source, and do not want Amazon Kendra to index all your database content after the first sync, you can choose to sync only new, modified, or deleted documents.
- As a best practice, provide Amazon Kendra with read-only database credentials.
- As a best practice, avoid adding tables with sensitive data or personal identifiable information (PII).

Microsoft Teams

Microsoft Teams is an enterprise collaboration tool for messaging, meetings and file sharing. If you are a Microsoft Teams user, you can use Amazon Kendra to index your Microsoft Teams data source.

You can connect Amazon Kendra to your Microsoft Teams data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Microsoft Teams data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more
- Notes

Supported features

- Field mappings

- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Microsoft Teams data source, make these changes in your Microsoft Teams and AWS accounts.

In Microsoft Teams, make sure you have:

- Created a Microsoft Teams account in Office 365.
- Noted your Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal or in your OAuth application.
- Configured an OAuth application in the Azure portal and noted the client ID and client secret or client credentials. See Microsoft tutorial and Registered app example for more information.

> **Note**
>
> When you create or register an app in the Azure portal, the secret ID represents the actual secret value. You must take note or save the actual secret value immediately when creating the secret and app. You can access your secret by selecting the name of your application in the Azure portal and then navigating to the menu option on certificates and secrets.
> You can access your client ID by selecting the name of your application in the Azure portal and then navigating to the overview page. The Application (client) ID is the client ID.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Added the necessary permissions. You can choose to add all permissions, or you can limit the scope by selecting fewer permissions based on which entities you'd like to crawl. The following table lists the application level permissions by corresponding entity:

Entity	Required Permissions for Data Sync	Required Permissions for Identity Sync
Channel Post	• ChannelMessage.Read.All • Group.Read.All • User.Read • User.Read.All	TeamMember.Read.All
Channel Attachment	• ChannelMessage.Read.All • Group.Read.All • User.Read • User.Read.All	TeamMember.Read.All
Channel Wiki	• Group.Read.All • User.Read • User.Read.All	TeamMember.Read.All
Chat Message	• Chat.Read.All • ChatMessage.Read.All • ChatMember.Read.All • User.Read • User.Read.All • Group.Read.All	TeamMember.Read.All
Meeting Chat	• Chat.Read.All • ChatMessage.Read • ChatMember.Read.All • User.Read • User.Read.All • Group.Read.All	TeamMember.Read.All

Entity	Required Permissions for Data Sync	Required Permissions for Identity Sync
Chat Attachment	• Chat.Read.All • ChatMessage.Read • ChatMember.Read.All • User.Read • User.Read.All • Group.Read.All	TeamMember.Read.All
Meeting File	• Chat.Read.All • ChatMessage.Read.All • ChatMember.Read.All • User.Read • User.Read.All • Group.Read.All • Files.Read.All	TeamMember.Read.All
Calendar Meeting	• Chat.Read.All • ChatMessage.Read.All • ChatMember.Read.All • User.Read • User.Read.All • Group.Read.All • Files.Read.All	TeamMember.Read.All
Meeting Notes	• User.Read • User.Read.All • Group.Read.All • Files.Read.All	TeamMember.Read.All

- Checked each document is unique in Microsoft Teams and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Microsoft Teams authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Microsoft Teams data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Microsoft Teams data source, you must provide the necessary details of your Microsoft Teams data source so that Amazon Kendra can access your data. If you have not yet configured Microsoft Teams for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to Microsoft Teams

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Microsoft Teams connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Microsoft Teams connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.
   b. (Optional) **Description**—Enter an optional description for your data source.
   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.
   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.
   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. **Tenant ID**—Enter your Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal or in your OAuth application.
   b. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
   c. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Microsoft Teams authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

i. Enter following information in the **Create an AWS Secrets Manager secret window**:

A. **Secret name**—A name for your secret. The prefix ‘AmazonKendra-Microsoft Teams-’ is automatically added to your secret name.

B. For **Client ID** and **Client secret**—Enter the authentication credentials configured in Microsoft Teams in the Azure portal.

ii. Save and add your secret.

d. **Payment model**—You can choose a licensing and payment model for your Microsoft Teams account. Model A payment models are restricted to licensing and payment models that require security compliance. Model B payment models are suitable for licensing and payment models that do not require security compliance.

e. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

f. **Identity crawler**—Specify whether to turn on Amazon Kendra’s identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra’s identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

g. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

h. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. **Sync contents—**Select the types of content to crawl. You can choose to crawl chat, teams, and calendar content.

b. **Additional configuration—**Specify certain calendar start and end dates, user emails, team names, and channel names, attachments, and OneNotes.

c. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

   - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
   - New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
   - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency—**Choose how often to sync your data source content and update your index.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. **Default data source fields—**Select from the Amazon Kendra generated default data source fields you want to map to your index.

   b. **Add field—**To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Microsoft Teams

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as MSTEAMS when you use the TemplateConfiguration JSON schema. Also specify the data source as TEMPLATE when you call the CreateDataSource API.
- **Tenant ID**—You can find your tenant ID in the Properties of your Azure Active Directory Portal or in your OAuth application.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - FORCED_FULL_CRAWL to freshly index all content, replacing existing content each time your data source syncs with your index.
  - FULL_CRAWL to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - CHANGE_LOG to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Microsoft Teams account. The secret is stored in a JSON structure with the following keys:

```
{
    "clientId": "client ID",
    "clientSecret": "client secret"
}
```

- **IAM role**—Specify RoleArn when you call CreateDataSource to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Microsoft Teams connector and Amazon Kendra. For more information, see IAM roles for Microsoft Teams data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Document/content types**—Specify whether to crawl chat messages and attachments, channel posts and attachments, channel wikis, calendar content, meeting chats and files and notes.
- **Calendar content**—Specify a start and end date-time to crawl calendar content.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain content in Microsoft Teams. You can include or exclude team names, channel names, file names and file types, user email, OneNote sections, and OneNote pages.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **Field mappings**—Choose to map your Microsoft Teams data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Microsoft Teams template schema.

Learn more

To learn more about integrating Amazon Kendra with your Microsoft Teams data source, see:

- Intelligently search your organization's Microsoft Teams data source with the Amazon Kendra connector for Microsoft Teams

Notes

- When Access Control Lists (ACLs) are enabled, the "Sync only new or modified content" option is not available due to Microsoft Teams API limitations. We recommend using "Full sync" or "New, modified, or deleted content sync" modes instead, or disable ACLs if you need to use this sync mode.

Microsoft Yammer

> **Note**
>
> Microsoft Yammer connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

Microsoft Yammer is an enterprise collaboration tool for messaging, meetings and file sharing. If you are a Microsoft Yammer user, you can use Amazon Kendra to index your Microsoft Yammer data source.

You can connect Amazon Kendra to your Microsoft Yammer data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Microsoft Yammer data source connector, see Troubleshooting data sources.

Supported features

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Microsoft Yammer data source, make these changes in your Microsoft Yammer and AWS accounts.

In Microsoft Yammer, make sure you have:

- Created a Microsoft Yammer administrative accountin Office 365.
- Noted your Microsoft Yammer user name and password.
- Noted your Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal or in your OAuth application.
- Configured an OAuth application in the Azure portal and noted the client ID and client secret or client credentials. See Microsoft tutorial and Registered app example for more information.

> **Note**
>
> When you create or register an app in the Azure portal, the secret ID represents the actual secret value. You must take note or save the actual secret value immediately when creating the secret and app. You can access your secret by selecting the name of your

application in the Azure portal and then navigating to the menu option on certificates and secrets.
You can access your client ID by selecting the name of your application in the Azure portal and then navigating to the overview page. The Application (client) ID is the client ID.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Checked each document is unique in Microsoft Yammer and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Microsoft Yammer authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you

re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Microsoft Yammer data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Microsoft Yammer data source, you must provide the necessary details of your Microsoft Yammer data source so that Amazon Kendra can access your data. If you have not yet configured Microsoft Yammer for Amazon Kendra, see [Prerequisites](#).

Console

To connect Amazon Kendra to Microsoft Yammer

1. Sign in to the AWS Management Console and open the [Amazon Kendra console](#).
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Microsoft Yammer connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Microsoft Yammer connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.
   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.
   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:
   a. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
   b. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Microsoft Yammer authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.
      i. Enter following information in the **Create an AWS Secrets Manager secret window**:
         A. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Microsoft Yammer-' is automatically added to your secret name.
         B. For **Username**, **Password**—Enter your Microsoft Yammer user name and password.
         C. For **Client ID**, **Client secret**—Enter the authentication credentials configured in Microsoft Yammer in the Azure portal.
      ii. Save and add your secret.
   c. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.
   d. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly

searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

e. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

f. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. **Since date**—Specify the date to begin crawling your data in Microsoft Yammer.

b. **Sync contents**—Select the type of content to crawl. For example, public message, private messages, and attachments.

c. **Additional configuration**—Specify certain community names you want to crawl, and also use regular expression patterns to include or exclude certain content.

d. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

e. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

f. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. **Default data source fields**—Select from the Amazon Kendra generated default data source fields you want to map to your index.

   b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Microsoft Yammer

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `YAMMER` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Microsoft Yammer account. The secret is stored in a JSON structure with the following keys:

```
{
    "username": "user name",
    "password": "password",
    "clientId": "client ID",
    "clientSecret": "client secret"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Microsoft Yammer connector and Amazon Kendra. For more information, see IAM roles for Microsoft Yammer data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Document/content types**—Specify whether to crawl community content, messages and attachments, and private messages.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain content.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **Field mappings**—Choose to map your Microsoft Yammer data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Microsoft Yammer template schema.

Learn more

To learn more about integrating Amazon Kendra with your Microsoft Yammer data source, see:

- Announcing the Yammer connector for Amazon Kendra

MySQL

> **Note**
>
> MySQL connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the

> Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

MySQL is an open source relational database management systen. If you are a MySQL user, you can use Amazon Kendra to index your MySQL data source. The Amazon Kendra MySQL data source connector supports MySQL 8.0. 21.

You can connect Amazon Kendra to your MySQL data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra MySQL data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Notes

Supported features

- Field mappings
- User context filtering
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your MySQL data source, make these changes in your MySQL and AWS accounts.

In MySQL, make sure you have:

- Noted your database user name and password.

> **Important**
>
> As a best practice, provide Amazon Kendra with read-only database credentials.

- Copied your database host url, port, and instance.
- Checked each document is unique in MySQL and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your MySQL authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your MySQL data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your MySQL data source you must provide details of your MySQL credentials so that Amazon Kendra can access your data. If you have not yet configured MySQL for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to MySQL

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **MySQL connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **MySQL connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:
   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.
   b. (Optional) **Description**—Enter an optional description for your data source.
   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.
   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.
   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:
   a. In **Source**, enter the following information:
   b. **Host**— Enter the database host name.

c. **Port**— Enter the database port.

d. **Instance**— Enter the database instance.

e. **Enable SSL certificate location**—Choose to enter the Amazon S3 path to your SSL certificate file.

f. In **Authentication**—enter the following information:

   - **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your MySQL authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

     A. Enter following information in the **Create an AWS Secrets Manager secret window**:

        I. **Secret name**—A name for your secret. The prefix 'AmazonKendra-MySQL-' is automatically added to your secret name.

        II. For **Database user name**, and **Password**—Enter the authentication credential values you copied from your database.

     B. Choose **Save**.

g. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. In **Sync scope**, choose from the following options :

   - **SQL query**—Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.

- **Primary key column**—Provide the primary key for the database table. This identifies a table within your database.
- **Title column**—Provide the name of the document title column within your database table.
- **Body column**—Provide the name of the document body column within your database table.

b. In **Additional configuration – *optional***, choose from the following options to sync specific content instead of syncing all files:

- **Change-detecting columns**—Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns.
- **Users' IDs column**—Enter the name of the column which contains User IDs to be allowed access to content.
- **Groups column**—Enter the name of the column that contains groups to be allowed access to content.
- **Source URLs column**—Enter the name of the column which contains Source URLs to be indexed.
- **Time stamps column**—Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
- **Time zones column**—Enter the name of the column which contains time zones for the content to be crawled.
- **Time stamps format**—Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

c. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.

- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency**—How often Amazon Kendra will sync with your data source.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. Select from the generated default data source fields—**Document IDs**, **Document titles**, and **Source URLs**—you want to map to Amazon Kendra index.

   b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to MySQL

You must specify the following using the [TemplateConfiguration](#) API:

- **Data source**—Specify the data source type as `JDBC` when you use the [TemplateConfiguration](#) JSON schema. Also specify the data source as `TEMPLATE` when you call the [CreateDataSource](#) API.
- **Database type**—You must specify the database type as `mySql`.
- **SQL query**—Specify SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.

- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials you created in your MySQL account. The secret is stored in a JSON structure with the following keys:

```
{
    "user name": "database user name",
    "password": "password"
}
```

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the MySQL connector and Amazon Kendra. For more information, see [IAM roles for MySQL data sources](#).

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include specific content using user IDs, groups, source URLs, time stamps, and time zones.
- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your MySQL data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

Notes

- Deleted database rows will not be tracked in when Amazon Kendra checks for updated content.
- The size of field names and values in a row of your database can't exceed 400KB.
- If you have a large amount of data in your database data source, and do not want Amazon Kendra to index all your database content after the first sync, you can choose to sync only new, modified, or deleted documents.
- As a best practice, provide Amazon Kendra with read-only database credentials.
- As a best practice, avoid adding tables with sensitive data or personal identifiable information (PII).

Oracle Database

> **Note**
>
> Oracle Database connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

Oracle Database is a database management system. If you are a Oracle Database user, you can use Amazon Kendra to index your Oracle Database data source. The Amazon Kendra Oracle Database data source connector supports Oracle Database 18c, 19c, and 21c.

You can connect Amazon Kendra to your Oracle Database data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Oracle Database data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Notes

Supported features

- Field mappings
- User context filtering
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Oracle Database data source, make these changes in your Oracle Database and AWS accounts.

In Oracle Database, make sure you have:

- Noted your database user name and password.

> **Important**
>
> As a best practice, provide Amazon Kendra with read-only database credentials.

- Copied your database host url, port, and instance.
- Checked each document is unique in Oracle Database and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Oracle Database authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Oracle Database data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Oracle Database data source you must provide details of your Oracle Database credentials so that Amazon Kendra can access your data. If you have not yet configured Oracle Database for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Oracle Database

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Oracle Database connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Oracle Database connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

    a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

    b. (Optional) **Description**—Enter an optional description for your data source.

    c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

    d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. In **Source**, enter the following information:

   b. **Host**— Enter the database host name.

   c. **Port**— Enter the database port.

   d. **Instance**— Enter the database instance.

   e. **Enable SSL certificate location**—Choose to enter the Amazon S3 path to your SSL certificate file.

   f. In **Authentication**—enter the following information:

      - **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Oracle Database authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

        A. Enter following information in the **Create an AWS Secrets Manager secret window**:

           I. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Oracle Database-' is automatically added to your secret name.

           II. For **Database user name**, and **Password**—Enter the authentication credential values you copied from your database.

        B. Choose **Save**.

   g. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

   h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

      > **Note**
      >
      > IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

   i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. In **Sync scope**, choose from the following options :

   - **SQL query—**Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
   - **Primary key column—**Provide the primary key for the database table. This identifies a table within your database.
   - **Title column—**Provide the name of the document title column within your database table.
   - **Body column—**Provide the name of the document body column within your database table.

b. In **Additional configuration – *optional***, choose from the following options to sync specific content instead of syncing all files:

   - **Change-detecting columns—**Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns.
   - **User IDs column—**Enter the name of the column which contains User IDs to be allowed access to content.
   - **Groups column—**Enter the name of the column that contains groups to be allowed access to content.
   - **Source URLs column—**Enter the name of the column which contains Source URLs to be indexed.
   - **Time stamps column—**Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
   - **Time zones column—**Enter the name of the column which contains time zones for the content to be crawled.
   - **Time stamps format—**Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

c. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your

data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency**—How often Amazon Kendra will sync with your data source.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

a. Select from the generated default data source fields—**Document IDs**, **Document titles**, and **Source URLs**—you want to map to Amazon Kendra index.

b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Oracle Database

You must specify the following using the TemplateConfiguration API:

- **Data source**—Specify the data source type as `JDBC` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.

- **Database type**—You must specify the database type as `oracle`.
- **SQL query**—Specify SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials you created in your Oracle Database account. The secret is stored in a JSON structure with the following keys:

```
{
    "user name": "database user name",
    "password": "password"
}
```

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Oracle Database connector and Amazon Kendra. For more information, see IAM roles for Oracle Database data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include specific content using user IDs, groups, source URLs, time stamps, and time zones.
- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Oracle Database data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Oracle Database template schema.

Notes

- Deleted database rows will not be tracked in when Amazon Kendra checks for updated content.
- The size of field names and values in a row of your database can't exceed 400KB.
- If you have a large amount of data in your database data source, and do not want Amazon Kendra to index all your database content after the first sync, you can choose to sync only new, modified, or deleted documents.
- As a best practice, provide Amazon Kendra with read-only database credentials.

- As a best practice, avoid adding tables with sensitive data or personal identifiable information (PII).

PostgreSQL

> **Note**
>
> PostgreSQL connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

PostgreSQL is an open source database management system. If you are a PostgreSQL user, you can use Amazon Kendra to index your PostgreSQL data source. The Amazon Kendra PostgreSQL data source connector supports PostgreSQL 9.6.

You can connect Amazon Kendra to your PostgreSQL data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra PostgreSQL data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Notes

Supported features

- Field mappings
- User context filtering
- Inclusion/exclusion filters

- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your PostgreSQL data source, make these changes in your PostgreSQL and AWS accounts.

In PostgreSQL, make sure you have:

- Noted your database user name and password.

> **Important**
>
> As a best practice, provide Amazon Kendra with read-only database credentials.

- Copied your database host url, port, and instance.
- Checked each document is unique in PostgreSQL and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your PostgreSQL authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your PostgreSQL data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your PostgreSQL data source you must provide details of your PostgreSQL credentials so that Amazon Kendra can access your data. If you have not yet configured PostgreSQL for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to PostgreSQL

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **PostgreSQL connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **PostgreSQL connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

a. In **Name and description**, for **Data source name—**Enter a name for your data source. You can include hyphens but not spaces.

b. (Optional) **Description—**Enter an optional description for your data source.

c. In **Default language—**Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag—**Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

   a. In **Source**, enter the following information:

   b. **Host—** Enter the database host name.

   c. **Port—** Enter the database port.

   d. **Instance—** Enter the database instance.

   e. **Enable SSL certificate location—**Choose to enter the Amazon S3 path to your SSL certificate file.

   f. In **Authentication—**enter the following information:

      - **AWS Secrets Manager secret—**Choose an existing secret or create a new Secrets Manager secret to store your PostgreSQL authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

        A. Enter following information in the **Create an AWS Secrets Manager secret window**:

           I. **Secret name—**A name for your secret. The prefix ‘AmazonKendra-PostgreSQL-’ is automatically added to your secret name.

           II. For **Database user name**, and **Password—**Enter the authentication credential values you copied from your database.

        B. Choose **Save**.

   g. **Virtual Private Cloud (VPC)—**You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. In **Sync scope**, choose from the following options :

- **SQL query**—Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
- **Primary key column**—Provide the primary key for the database table. This identifies a table within your database.
- **Title column**—Provide the name of the document title column within your database table.
- **Body column**—Provide the name of the document body column within your database table.

b. In **Additional configuration – *optional***, choose from the following options to sync specific content instead of syncing all files:

- **Change-detecting columns**—Enter the names of the columns that Amazon Kendra will use to detect content changes. Amazon Kendra will re-index content when there is a change in any of these columns.
- **Users' IDs column**—Enter the name of the column which contains User IDs to be allowed access to content.
- **Groups column**—Enter the name of the column that contains groups to be allowed access to content.
- **Source URLs column**—Enter the name of the column which contains Source URLs to be indexed.

- **Time stamps column**—Enter the name of the column which contains time stamps. Amazon Kendra uses time stamp information to detect changes in your content and sync only changed content.
- **Time zones column**—Enter the name of the column which contains time zones for the content to be crawled.
- **Time stamps format**—Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.

c. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

d. In **Sync run schedule**, for **Frequency**—How often Amazon Kendra will sync with your data source.

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. Select from the generated default data source fields—**Document IDs**, **Document titles**, and **Source URLs**—you want to map to Amazon Kendra index.

   b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this

page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to PostgreSQL

You must specify the following using the TemplateConfiguration API:

- **Data source**—Specify the data source type as `JDBC` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Database type**—You must specify the database type as `postgresql`.
- **SQL query**—Specify SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Kendra will crawl all database content that matches your query.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials you created in your PostgreSQL account. The secret is stored in a JSON structure with the following keys:

```
{
    "user name": "database user name",
    "password": "password"
```

```
}
```

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the PostgreSQL connector and Amazon Kendra. For more information, see IAM roles for PostgreSQL data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include specific content using user IDs, groups, source URLs, time stamps, and time zones.
- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your PostgreSQL data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see PostgreSQL template schema.

Notes

- Deleted database rows will not be tracked in when Amazon Kendra checks for updated content.
- The size of field names and values in a row of your database can't exceed 400KB.
- If you have a large amount of data in your database data source, and do not want Amazon Kendra to index all your database content after the first sync, you can choose to sync only new, modified, or deleted documents.
- As a best practice, provide Amazon Kendra with read-only database credentials.
- As a best practice, avoid adding tables with sensitive data or personal identifiable information (PII).

Quip

Note

Quip connector remains fully supported for existing customers through May 31, 2026. While this connector is no longer available for new users, current users can continue to use it without interruption. We are continuously evolving our connector portfolio to offer more scalable and customizable solutions. For future integrations, we recommend exploring the Amazon Kendra Custom Connector Framework[1], designed to support a broader range of enterprise use cases with enhanced flexibility.

Quip is a collaborative productivity software that offers real time document-authoring capabilities. You can use Amazon Kendra to index your Quip folders, files, file comments, chatrooms, and attachments.

You can connect Amazon Kendra to your Quip data source using the Amazon Kendra console and the QuipConfiguration API.

For troubleshooting your Amazon Kendra Quip data source connector, see Troubleshooting data sources.

Topics

- Supported features

- Prerequisites
- Connection instructions
- Learn more

Supported features

Amazon Kendra Quip data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Quip data source, make these changes in your Quip and AWS accounts.

In Quip, make sure you have:

- A Quip account with administrative permissions.
- Created Quip authentication credentials that include a personal access token. The token is used as your authentication credential stored in an AWS Secrets Manager secret. See Quip documentation on authentication for more information.

  > **Note**
  >
  > We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Copied your Quip site domain. For example, *https://quip-company.quipdomain.com/browse* where *quipdomain* is the domain.
- Checked each document is unique in Quip and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same

document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Quip authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don’t have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Quip data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Quip data source, you must provide the necessary details of your Quip data source so that Amazon Kendra can access your data. If you have not yet configured Quip for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to Quip

1. Sign in to the AWS Management Console and open the [Amazon Kendra console](#).
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Quip connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Quip connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

    a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

    b. (Optional) **Description**—Enter an optional description for your data source.

    c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

    d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

    e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

    a. **Quip domain name**—Enter the Quip you copied from your Quip account.

    b. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Quip authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

        i. Enter following information in the **Create an AWS Secrets Manager secret window**:

            A. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Quip-' is automatically added to your secret name.

            B. **Quip token**—Enter the Quip personal access configured Quip.

ii. Add and save your secret.

c. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

d. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

e. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. **Add Quip folder IDs to crawl**—The Quip folder IDs you want to crawl.

> **Note**
>
> To crawl a root folder, including all sub-folders and documents inside it, add the root folder ID. To crawl specific sub-folders, add the specific sub-folder IDs.

b. **Additional configuration (content types)**—Enter the content types you want to crawl.

c. **Regex patterns**—Regular expression patterns to include or exclude certain files. You can add up to 100 patterns.

d. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index

e. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

a. Select from the generated default data source fields you want to map to Amazon Kendra index.

b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Quip

You must specify the following using QuipConfiguration API:

- **Quip site domain**—For example, *https://quip-company.quipdomain.com/browse* where *quipdomain* is the domain.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Quip account. The secret is stored in a JSON structure with the following keys:

```
{
    "accessToken": "token"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Quip connector and Amazon Kendra. For more information, see IAM roles for Quip data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` as part of the data source configuration. See Configuring Amazon Kendra to use a VPC.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain files.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the

inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Folders**—Specify Quip folders and subfolders you want to index

> **Note**
>
> To crawl a root folder, including all sub-folders and documents inside it, input the root folder ID. To crawl specific sub-folders, add the specific sub-folder IDs.

- **Attachments, Chat rooms, file comments**—Choose whether to include crawling of attachments, chat rooms content, and file comments.
- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Quip data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

Learn more

To learn more about integrating Amazon Kendra with your Quip data source, see:

- Search for knowledge in Quip documents with intelligent search using the Quip connector for Amazon Kendra

Salesforce

Salesforce is a customer relationship management (CRM) tool for managing support, sales, and marketing teams. You can use Amazon Kendra to index your Salesforce standard objects and even custom objects.

You can connect Amazon Kendra to your Salesforce data source using either the Amazon Kendra console, the TemplateConfiguration API, or the SalesforceConfiguration API.

Amazon Kendra has two versions of the Salesforce connector. Supported features of each version include:

Salesforce connector V1.0 / SalesforceConfiguration API

- Field mappings
- User access control
- Inclusion/exclusion filters

Salesforce connector V2.0 / TemplateConfiguration API

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

> **Note**
>
> Salesforce connector V1.0 / SalesforceConfiguration API ended in 2023. We recommend migrating to or using Salesforce connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra Salesforce data source connector, see Troubleshooting data sources.

Topics

- [Salesforce connector V2.0]

Salesforce connector V1.0

Salesforce is a customer relationship management (CRM) tool for managing support, sales, and marketing teams. You can use Amazon Kendra to index your Salesforce standard objects and even custom objects.

> **Important**
>
> Amazon Kendra uses the Salesforce API version 48. The Salesforce API limits the number of requests that you can make per day. If Salesforce exceeds those requests, it retries until it is able to continue.

> **Note**
>
> Salesforce connector V1.0 / SalesforceConfiguration API ended in 2023. We recommend migrating to or using Salesforce connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra Salesforce data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions

Supported features

Amazon Kendra Salesforce data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters

Prerequisites

Before you can use Amazon Kendra to index your Salesforce data source, make these changes in your Salesforce and AWS accounts.

In Salesforce, make sure you have:

- Created a Salesforce account and have noted the user name and password you use to connect to Salesforce.
- Created a Salesforce Connected App account with OAuth activated and have copied the consumer key (client ID) and consumer secret (client secret) assigned to your Salesforce Connected App. The client ID and client secret are used as your authentication credentials stored in an AWS Secrets Manager secret. See Salesforce documentation on Connected Apps for more information.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Copied the Salesforce security token associated with the account used to connect to Salesforce.
- Copied the URL of the Salesforce instance that you want to index. Typically, this is `https://<company>.salesforce.com/`. The server must be running a Salesforce connected app.
- Added credentials to your Salesforce server for a user with read-only access to Salesforce by cloning the ReadOnly profile and then adding the View All Data and Manage Articles permissions. These credentials identify the user making the connection and the Salesforce connected app that Amazon Kendra connects to.
- Checked each document is unique in Salesforce and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.

- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Salesforce authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Salesforce data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Salesforce data source, you must provide the necessary details of your Salesforce data source so that Amazon Kendra can access your data. If you have not yet configured Salesforce for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Salesforce

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Salesforce connector V1.0**, and then choose **Add connector**.
5. On the **Specify data source details** page, enter the following information:

    a. **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

    b. (Optional) **Description**—Enter an optional description for your data source.

    c. **Default language**— A language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in metadata overrides selected language.

    d. **Add new tag**—Tags to search and filter your resources or track your shared costs.

    e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

    a. **Salesforce URL**—Enter the instance URL for the Salesforce site that you want to index.

    b. For **Type of authentication**, choose between **Existing** and **New** to store your Salesforce authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

    - Enter following information in the **Create an AWS Secrets Manager secret window**:

        A. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Salesforce-' is automatically added to your secret name.

        B. For **User name**, **Password**, **Security token**, **Consumer key**, **Consumer secret**, and **Authentication URL**—Enter the authentication credential values you created in your Salesforce account.

        C. Choose **Save authentication**.

c. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

d. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. For **Crawl attachments**—Select to crawl all attached objects, articles, and feeds.

b. For **Standard objects**, **Knowledge articles**, and **Chatter feeds**—Select Salesforce entities or content types you want to crawl.

> **Note**
>
> You must provide configuration information for indexing at least one of standard objects, knowledge articles, or chatter feeds. If you choose to crawl **Knowledge articles** you must specify the types of knowledge articles to index, the name of the articles, and whether to index the standard fields of all knowledge articles or only the fields of a custom article type. If you choose to index custom articles, you must specify the internal name of the article type. You can specify upto 10 article types.

c. **Frequency**—How often Amazon Kendra will sync with your data source.

d. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

a. For **Standard knowledge article**, **Standard object attachments**, and **Additional suggested field mappings** —Select from the Amazon Kendra generated default data source fields you want to map to your index.

> **Note**
>
> An index mapping to `_document_body` is required. You can't change the mapping between the `Salesforce ID` field and the Amazon Kendra `_document_id` field.

b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Salesforce

You must specify the following the SalesforceConfiguration API:

- **Server URL**—The instance URL for the Salesforce site that you want to index.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Salesforce account. The secret is stored in a JSON structure with the following keys:

```
{
    "authenticationUrl": "OAUTH endpoint that Amazon Kendra connects to get an OAUTH token",
    "consumerKey": "Application public key generated when you created your Salesforce application",
    "consumerSecret": "Application private key generated when you created your Salesforce application.",
    "password": "Password associated with the user logging in to the Salesforce instance",
    "securityToken": "Token associated with the user account logging in to the Salesforce instance",
    "username": "User name of the user logging in to the Salesforce instance"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Salesforce connector and Amazon Kendra. For more information, see IAM roles for Salesforce data sources.
- You must provide configuration information for indexing at least one of standard objects, knowledge articles, or chatter feeds.
  - **Standard objects**—If you choose to crawl **Standard objects**, you must specify the name of the standard object and the name of the field in the standard object table that contains the document contents.
  - **Knowledge articles**—If you choose to crawl **Knowledge articles**, you must specify the types of knowledge articles to index, the states of the knowledge articles to index, and whether to index the standard fields of all knowledge articles or only the fields of a custom article type.
  - **Chatter feeds**—If you choose to crawl **Chatter feeds**, you must specify the name of the column in the Salesforce FeedItem table that contains the content to index.

You can also add the following optional features:

- **Inclusion and exclusion filters**—Specify whether to include or exclude certain file attachments.

  > **Note**
  >
  > Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Field mappings**—Choose to map your Salesforce data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

  > **Note**
  >
  > The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must

map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

Salesforce connector V2.0

Salesforce is a customer relationship management (CRM) tool for managing support, sales, and marketing teams. You can use Amazon Kendra to index your Salesforce standard objects and even custom objects.

The Amazon Kendra Salesforce data source connector supports the following Salesforce editions: Developer Edition and Enterprise Edition.

> **Note**
>
> Salesforce connector V1.0 / SalesforceConfiguration API ended in 2023. We recommend migrating to or using Salesforce connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra Salesforce data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more
- Notes

Supported features

Amazon Kendra Salesforce data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Salesforce data source, make these changes in your Salesforce and AWS accounts.

In Salesforce, make sure you have:

- Created a Salesforce administrative account and have noted the user name and password you use to connect to Salesforce.
- Copied the Salesforce security token associated with the account used to connect to Salesforce.
- Created a Salesforce Connected App account with OAuth activated and have copied the consumer key (client ID) and consumer secret (client secret) assigned to your Salesforce Connected App. The client ID and client secret are used as your authentication credentials stored in an AWS Secrets Manager secret. See Salesforce documentation on Connected Apps for more information.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Copied the URL of the Salesforce instance that you want to index. Typically, this is `https://<company>.salesforce.com/`. The server must be running a Salesforce connected app.
- Added credentials to your Salesforce server for a user with read-only access to Salesforce by cloning the ReadOnly profile and then adding the View All Data and Manage Articles permissions. These credentials identify the user making the connection and the Salesforce connected app that Amazon Kendra connects to.

- Checked each document is unique in Salesforce and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Salesforce authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Salesforce data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Salesforce data source, you must provide the necessary details of your Salesforce data source so that Amazon Kendra can access your data. If you have not yet configured Salesforce for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to Salesforce:

1. Sign in to the AWS Management Console and open the [Amazon Kendra console](https://console.aws.amazon.com/kendra/).
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Salesforce connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Salesforce connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. **Salesforce URL**—Enter The instance URL for the Salesforce site that you want to index.

   b. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see [User context filtering](https://docs.aws.amazon.com/kendra/latest/dg/user-context-filter.html).

c. Enter an existing secret or if you create a new secret, an AWS Secrets Manager secret window opens.

- **Authentication**—Enter following information in the **Create an AWS Secrets Manager secret window**:

  A. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Salesforce-' is automatically added to your secret name.

  B. For **User name**, **Password**, **Security token**, **Consumer key**, **Consumer secret**, and **Authentication URL**—Enter the authentication credential values you generated and downloaded from your Salesforce account.

  > **Note**
  >
  > If you use Salesforce Developer Edition, use `https://login.salesforce.com/services/oauth2/token` or the My Domain login URL (for example, *`https://MyCompany.my.salesforce.com`*) as the **Authentication URL**. If you use Salesforce Sandbox Edition, use `https://test.salesforce.com/services/oauth2/token` or the My Domain login URL (for example, *`MyDomainName--SandboxName.sandbox.my.salesforce.com`*) as the **Authentication URL**.

  C. Choose **Save authentication**.

d. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

e. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

f. **IAM role—**Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

g. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. For **Crawl attachments—**Select to crawl all attached Salesforce objects.

   b. For **Standard objects**, **Standard objects with attachments**, and **Standard object without attachment** and **Knowledge Articles—**Select Salesforce entities or content types you want to crawl.

   c. You must provide configuration information for indexing at least one of standard objects, knowledge articles, or chatter feeds. If you choose to crawl **Knowledge articles** you must specify the types of knowledge articles to index. You can choose published, archived, drafts and attachments.

   Regex filter—Specify a regex pattern to include specific catalog items.

8. For **Additional configuration**:

   - **ACL information** All access control lists are included by default. Deselecting an access control list will make all files in that category public.
   - **Regex patterns—**Add regular expression patterns to include or exclude certain files. You can add up to 100 patterns.

   Sync mode—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

   - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.

- New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

9. Choose **Next**.
10. On the **Set field mappings** page, enter the following information:

    a. For **Standard knowledge article**, **Standard object attachments**, and **Additional suggested field mappings** —Select from the Amazon Kendra generated default data source fields you want to map to your index.

    > **Note**
    >
    > An index mapping to `_document_body` is required. You can't change the mapping between the `Salesforce ID` field and the Amazon Kendra `_document_id` field. You can map any Salesforce field to the document title or document body Amazon Kendra reserved/default index fields.
    > If you map any Salesforce field to Amazon Kendra document title and document body fields, Amazon Kendra will use data from the document title and body fields in search responses.

    b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

    c. Choose **Next**.
11. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Salesforce

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `SALESFORCEV2` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Host URL**—Specify the Salesforce instance host URL.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Salesforce account. The secret is stored in a JSON structure with the following keys:

```
{
    "authenticationUrl": "OAUTH endpoint that Amazon Kendra connects to get an OAUTH token",
    "consumerKey": "Application public key generated when you created your Salesforce application",
    "consumerSecret": "Application private key generated when you created your Salesforce application",
    "password": "Password associated with the user logging in to the Salesforce instance",
    "securityToken": "Token associated with the user account logging in to the Salesforce instance",
    "username": "User name of the user logging in to the Salesforce instance"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Salesforce connector and Amazon Kendra. For more information, see IAM roles for Salesforce data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Inclusion and exclusion filters**—You can specify whether to include or exclude certain documents, accounts, campaigns, cases, contacts, leads, opportunities, solutions, tasks, groups, chatters, and custom entity files.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **Field mappings**—Choose to map your Salesforce data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

> **Note**
>
> An index mapping to `_document_body` is required. You can't change the mapping between the `Salesforce ID` field and the Amazon Kendra `_document_id` field. You can map any Salesforce field to the document title or document body Amazon Kendra reserved/default index fields.
> If you map any Salesforce field to Amazon Kendra document title and document body fields, Amazon Kendra will use data from the document title and body fields in search responses.

For a list of other important JSON keys to configure, see Salesforce template schema.

Learn more

To learn more about integrating Amazon Kendra with your Salesforce data source, see:

- Announcing the updated Salesforce connector (V2) for Amazon Kendra

Notes

- When Access Control Lists (ACLs) are enabled, the "Sync only new or modified content" option is not available due to Salesforce API limitations. We recommend using "Full sync" or "New, modified, or deleted content sync" modes instead, or disable ACLs if you need to use this sync mode.

ServiceNow

ServiceNow provides a cloud-based service management system to create and manage organization-level workflows, such as IT services, ticketing systems, and support. You can use Amazon Kendra to index your ServiceNow catalogs, knowledge articles, incidents, and their attachments.

You can connect Amazon Kendra to your ServiceNow data source using either the Amazon Kendra console, the TemplateConfiguration API, or the ServiceNowConfiguration API.

Amazon Kendra has two versions of the ServiceNow connector. Supported features of each version include:

ServiceNow connector V1.0 / ServiceNowConfiguration API

- Field mappings
- ServiceNow instance versions: London, Others
- Inclusion/exclusion filters

ServiceNow connector V2.0 / TemplateConfiguration API

- Field mappings
- User access control
- Inclusion/exclusion filters
- Full and incremental content syncs
- ServiceNow instance versions: Rome, Sandiego, Tokyo, Others
- Virtual private cloud (VPC)

> **Note**
>
> ServiceNow connector V1.0 / ServiceNowConfiguration API ended in 2023. We recommend migrating to or using ServiceNow connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra ServiceNow data source connector, see Troubleshooting data sources.

Topics

- ServiceNow connector V1.0
- ServiceNow connector V2.0
- Specifying documents to index with a query

ServiceNow connector V1.0

ServiceNow provides a cloud-based service management system to create and manage organization-level workflows, such as IT services, ticketing systems, and support. You can use Amazon Kendra to index your ServiceNow catalogs, knowledge articles, and their attachments.

> **Note**
>
> ServiceNow connector V1.0 / ServiceNowConfiguration API ended in 2023. We recommend migrating to or using ServiceNow connector V2.0 / TemplateConfiguration API.

For troubleshooting your Amazon Kendra ServiceNow data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more

Supported features

Amazon Kendra ServiceNow data source connector supports the following features:

- ServiceNow instance versions: London, Others
- Inclusion/exclusion patterns: Service catalogs, knowledge articles, and their attachments

Prerequisites

Before you can use Amazon Kendra to index your ServiceNow data source, make these changes in your ServiceNow and AWS accounts.

In ServiceNow, make sure you have:

- Created a ServiceNow administrator account and have created a ServiceNow instance.
- Copied the host of your ServiceNow instance URL. For example, if the URL of the instance is *`https://your-domain.service-now.com`*, the format for the host URL you enter is *`your-domain.service-now.com`*.
- Noted your basic authentication credentials containing a user name and password to allow Amazon Kendra to connect to your ServiceNow instance.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **Optional:** Configured an OAuth 2.0 credential token that can identify Amazon Kendra and generate a user name, password, a client ID, and a client secret. The user name and password must provide access to the ServiceNow knowledge base and service catalog. See [ServiceNow documentation on OAuth 2.0 authentication] for more information.
- Added the following permissions:
  - kb_category
  - kb_knowledge
  - kb_knowledge_base
  - kb_uc_cannot_read_mtom
  - kb_uc_can_read_mtom
  - sc_catalog
  - sc_category
  - sc_cat_item
  - sys_attachment
  - sys_attachment_doc

  - sys_user_role
- Checked each document is unique in ServiceNow and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your ServiceNow authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your ServiceNow data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your ServiceNow data source, you must provide the necessary details of your ServiceNow data source so that Amazon Kendra can access your data. If you have not yet configured ServiceNow for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to ServiceNow

1. Sign in to the AWS Management Console and open the [Amazon Kendra console](https://console.aws.amazon.com/kendra/).
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

> **Note**
>
> You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **ServiceNow connector V1.0**, and then choose **Add data source**.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

   b. (Optional) **Description**—Enter an optional description for your data source.

   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. **ServiceNow host**—Enter the ServiceNow host URL.

   b. **ServiceNow version**—Select your ServiceNow version.

   c. Choose between **Basic authentication** and **Oauth 2.0 authentication** based on your use case.

   d. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your ServiceNow authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

   i. **Secret name**—A name for your secret. The prefix ‘AmazonKendra-ServiceNow-’ is automatically added to your secret name.

   ii. If using Basic Authentication—Enter the **Secret name**, **Username**, and **Password** for your ServiceNow account.

       If using OAuth2 Authentication—Enter the **Secret name**, **Username**, **Password**, **Client ID**, and **Client Secret** you created in your ServiceNow account.

   iii. Choose **Save and add secret**.

   e. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

   f. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. **Include knowledge articles**—Choose to index knowledge articles.

   b. **Type of knowledge articles**—Choose between **Include only public articles** and **Include articles based on ServiceNow filter query** based on your use case. If you select **Include articles based on ServiceNow filter query**, you must enter a **Filter query** copied from your ServiceNow account.

   c. **Include knowledge articles attachments**—Choose to index knowledge article attachments. You can also select specific file types to index.

   d. **Include catalog items**—Choose to index catalog items.

   e. **Include catalog item attachments**—Choose to index catalog item attachments. You can also select specific file types to index.

   f. **Frequency**—How often Amazon Kendra will sync with your data source.

   g. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

a. **Knowledge articles** and **Service catalog** —Select from the Amazon Kendra generated default data source fields and additional suggested field mappings that you want to map to your index.

b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to ServiceNow

You must specify the following using ServiceNowConfiguration API:

- **Data source URL**—Specify the ServiceNow URL. The host endpoint should look like the following: *your-domain.service-now.com*.
- **Data source host instance**—Specify the ServiceNow host instance version as either `LONDON` or `OTHERS`.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of a Secrets Manager secret that contains the authentication credentials you created in your ServiceNow account.

  If you are using basic authentication, the secret is stored in a JSON structure with the following keys:

  ```
  {
      "username": "user name",
      "password": "password"
  }
  ```

  If you are using OAuth2 authentication, the secret is stored in a JSON structure with the following keys:

  ```
  {
  ```

```
    "username": "user name",
    "password": "password",
    "clientId": "client id",
    "clientSecret": "client secret"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the ServiceNow connector and Amazon Kendra. For more information, see IAM roles for ServiceNow data sources.

You can also add the following optional features:

- **Field mappings**—Choose to map your ServiceNow data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

- **Inclusion and exclusion filters**—Specify whether to include or exclude certain file attachments of catalogs and knowledge articles.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Indexing parameters**—You can also choose to specify whether to:
  - Index knowledge articles and service catalogs, or both of these. If you choose to index knowledge articles and service catalog items, you must provide the name of the

ServiceNow field that is mapped to the index document contents field in the Amazon Kendra index.

- Index attachments to knowledge articles and catalog items.
- Use a ServiceNow query that selects documents from one or more knowledge bases. The knowledge bases can be public or private. For more information, see Specifying documents to index with a query.

Learn more

To learn more about integrating Amazon Kendra with your ServiceNow data source, see:

- Getting started with Amazon Kendra ServiceNow Online connector

ServiceNow connector V2.0

ServiceNow provides a cloud-based service management system to create and manage organization-level workflows, such as IT services, ticketing systems, and support. You can use Amazon Kendra to index your ServiceNow catalogs, knowledge articles, incidents, and their attachments.

For troubleshooting your Amazon Kendra ServiceNow data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more

Supported features

Amazon Kendra ServiceNow data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters

- Full and incremental content syncs
- ServiceNow instance versions: Rome, Sandiego, Tokyo, Others
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your ServiceNow data source, make these changes in your ServiceNow and AWS accounts.

In ServiceNow, make sure you have:

- Created a Personal or Enterprise Developer Instance and have a ServiceNow instance with an administrative role.
- Copied the host of your ServiceNow instance URL. The format for the host URL you enter is *`your-domain.service-now.com`*. You need your ServiceNow instance URL to connect to Amazon Kendra.
- Noted your basic authentication credentials of a user name and password to allow Amazon Kendra to connect to your ServiceNow instance.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- **Optional:** Configured OAuth 2.0 client credentials that can identify Amazon Kendra using a user name, password, and a generated client ID, and a client secret. See [ServiceNow documentation on OAuth 2.0 authentication] for more information.
- Added the following permissions:
  - kb_category
  - kb_knowledge
  - kb_knowledge_base
  - kb_uc_cannot_read_mtom
  - kb_uc_can_read_mtom
  - sc_catalog

  - sc_category
  - sc_cat_item
  - sys_attachment
  - sys_attachment_doc
  - sys_user_role
- Checked each document is unique in ServiceNow and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your ServiceNow authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your ServiceNow data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your ServiceNow data source, you must provide the necessary details of your ServiceNow data source so that Amazon Kendra can access your data. If you have not yet configured ServiceNow for Amazon Kendra see Prerequisites.

Console

To connect Amazon Kendra to ServiceNow

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **ServiceNow connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **ServiceNow connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.
   b. (Optional) **Description**—Enter an optional description for your data source.
   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.
   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.
   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:

   a. **ServiceNow host**—Enter the ServiceNow host URL. The format for the host URL you enter is *`your-domain.service-now.com`*.

b. **ServiceNow version**—Select your ServiceNow instance version. You can select from Rome, Sandiego, Tokyo, or Others.

c. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.

d. **Authentication**—Choose between **Basic authentication** and **Oauth 2.0 authentication**.

e. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your ServiceNow authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens. Enter the following information in the window:

   i. **Secret name**—A name for your secret. The prefix ‘AmazonKendra-ServiceNow-’ is automatically added to your secret name.

   ii. If using Basic Authentication—Enter the **Secret name**, **Username**, and **Password** for your ServiceNow account.

   If using OAuth2.0 Authentication—Enter the **Secret name**, **Username**, **Password**, **Client ID**, and **Client Secret** you created in your ServiceNow account.

   iii. Save and add your secret.

f. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

g. **Identity crawler**—Specify whether to turn on Amazon Kendra’s identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra’s identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

h. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

i. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. For **Knowledge articles**, choose from the following options :

   - **Knowledge articles**—Choose to index knowledge articles.
   - **Knowledge article attachments**—Choose to index knowledge article attachments.
   - **Type of knowledge articles**—Choose between **Only public articles** and **Knowledge articles based on ServiceNow filter query** based on your use case. If you select **Include articles based on ServiceNow filter query**, you must enter a **Filter query** copied from your ServiceNow account. Example filter queries include: *workflow_state=draft^EQ*, *kb_knowledge_base=dfc19531bf2021003f07e2c1ac0739ab^text ISNOTEMPTY^EQ*, *article_type=text^active=true^EQ*.

     > **Important**
     >
     > If you choose to crawl **Only public articles**, Amazon Kendra crawls only knowledge articles assigned a public access role in ServiceNow.

   - **Include articles based on short description filter**—Specify regular expression patterns to include or exclude specific articles.

   b. For **Service catalog items**:

   - **Service catalog items**—Choose to index service catalog items.
   - **Service catalog item attachments**—Choose to index service catalog item attachments.
   - **Active service catalog items**—Choose to index active service catalog items.
   - **Inactive service catalog items**—Choose to index inactive service catalog items.

- **Filter query—**Choose to include service catalog items based on a filter defined in your ServiceNow instance. Example filter queries include: *short_descriptionLIKEAccess^category=2809952237b1300054b6a3549dbe5dd4* *nameSTARTSWITHService^active=true^EQ*.
- **Include service catalog items based on short description filter—**Specify a regex pattern to include specific catalog items.

c. For **Incidents**:

- **Incidents—**Choose to index service incidents.
- **Incident attachments—**Choose to index incident attachments.
- **Active incidents—**Choose to index active incidents.
- **Inactive incidents—**Choose to index inactive incidents.
- **Active incident type—**Choose between **All incidents**, **Open incidents**, **Open - unassigned incidents**, and **Resolved incidents** depending on your use case.
- **Filter query—**Choose to include incidents based on a filter defined in your ServiceNow instance. Example filter queries include: *short_descriptionLIKETest^urgency=3^state=1^EQ*, *priority=2^category=software^EQ* .
- **Include incidents based on short description filter—**Specify a regex pattern to include specific incidents.

d. For **Additional configuration**:

- **ACL information—**Access control lists for entities you have selected are included by default. Deselecting an access control list will make all files in that category public. ACL options are automatically deactivated for entities not selected. For public articles ACL is not applied.
- For **Maximum file size –** Specify the file size limit in MBs that Amazon Kendra will crawl. Amazon Kendra will crawl only the files within the size limit you define. The default file size is 50MB. The maximum file size should be greater than 0MB and less than or equal to 50MB.
- **Attachment regex patterns—**Add regular expression patterns to include or exclude certain attached files of catalogs, knowledge articles, and incidents. You can add up to 100 patterns.

e. **Sync mode—**Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

   - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
   - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

f. In **Sync run schedule**, for **Frequency—**Choose how often to sync your data source content and update your index.

g. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

   a. **Default field mappings—**Select from the Amazon Kendra generated default data source fields that you want to map to your index.

   b. **Add field—**To add custom data source fields to create an index field name to map to and the field data type.

   c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to ServiceNow

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `SERVICENOWV2` when you use the [TemplateConfiguration](#) JSON schema. Also specify the data source as `TEMPLATE` when you call the [CreateDataSource](#) API.
- **Host URL**—Specify the ServiceNow host instance version. For example, *`your-domain.service-now.com`*.
- **Authentication type**—Specify the type of authentication you use, whether `basicAuth` or `OAuth2` for your ServiceNow instance.
- **ServiceNow instance version**—Specify the ServiceNow instance you use, whether `Tokyo`, `Sandiego`, `Rome`, or `Others`.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of a Secrets Manager secret that contains the authentication credentials you created in your ServiceNow account.

  If you use basic authentication, the secret is stored in a JSON structure with the following keys:

```
{
    "username": "user name",
    "password": "password"
}
```

- If you use OAuth2 client credentials, the secret is stored in a JSON structure with the following keys:

```
{
    "username": "user name",
    "password": "password",
```

```
    "clientId": "client id",
    "clientSecret": "client secret"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the ServiceNow connector and Amazon Kendra. For more information, see [IAM roles for ServiceNow data sources](#).

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see [Configuring Amazon Kendra to use an Amazon VPC](#).
- **Inclusion and exclusion filters**—You can specify whether to include or exclude certain attached files using the file names and the file types of knowledge articles, service catalogs, and incidents.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Specific documents to index**—You can use a ServiceNow query to specify the documents you want from one or more knowledge bases, including private knowledge bases. Access to the knowledge bases is determined by the user that you use to connect to the ServiceNow instance. For more information, see [Specifying documents to index with a query](#).
- **Indexing parameters**—You can also choose to specify whether to:
  - Index knowledge articles, service catalogs, and incidents or all of these. If you choose to index knowledge articles, service catalog items and incidents, you must provide the name of the ServiceNow field that is mapped to the index document contents field in the Amazon Kendra index.
  - Index attachments to knowledge articles, service catalog items and incidents.

    - Include knowledge articles, service catalog items and incidents based on the `short description` filter pattern.
    - Choose to filter active and inactive service catalog items and incidents.
    - Choose to filter incidents based on incident type.
    - Choose which entities should have their ACL crawled.
    - You can use a ServiceNow query to specify the documents you want from one or more knowledge bases, including private knowledge bases. Access to the knowledge bases is determined by the user that you use to connect to the ServiceNow instance. For more information, see Specifying documents to index with a query.
- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **Field mappings**—Choose to map your ServiceNow data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see ServiceNow template schema.

Learn more

To learn more about integrating Amazon Kendra with your ServiceNow data source, see:

- Getting started with Amazon KendraAnnouncing the updated ServiceNow connector (V2) for Amazon Kendra

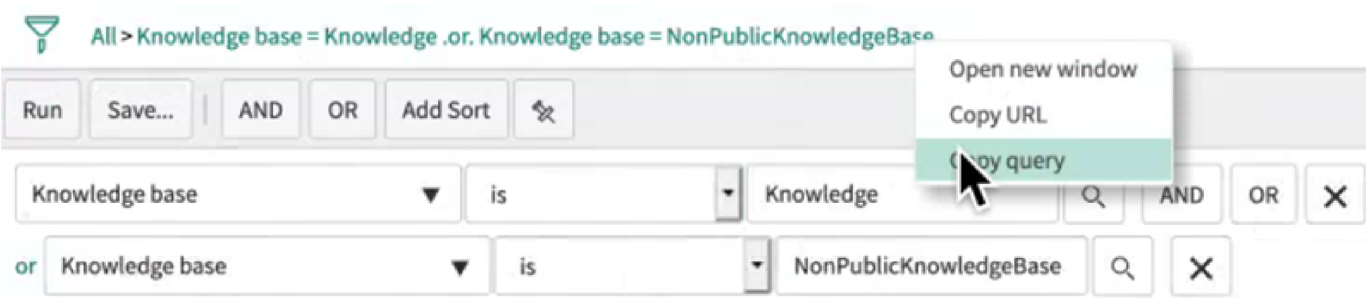
Specifying documents to index with a query

You can use a ServiceNow query to specify the documents you want to include in an Amazon Kendra index. When you use a query, you can specify multiple knowledge bases, including private knowledge bases. Access to the knowledge bases is determined by the user that you use to connect to the ServiceNow instance.

To build a query, you use the ServiceNow query builder. You can use the builder to create the query and to test that the query returns the correct list of documents.

To create a query using the ServiceNow console

1. Log in to the ServiceNow console.
2. From the left menu, choose **Knowledge**, then **Articles**, and the choose **All**.
3. At the top of the page, choose the filter icon.
4. Use the query builder to create the query.
5. When the query is complete, right click the query and choose **Copy query** to copy the query from the query builder. Save this query to use in Amazon Kendra.



Make sure that you don't change any query parameter when you copy the query. If any of the query parameters are not recognized, ServiceNow treats the parameter as empty and doesn't use it to filter the results.

Slack

Slack is an enterprise communications app that lets users send messages and attachments through various public and private channels. You can use Amazon Kendra to index your Slack public and private channels, bot and archive messages, files and attachments, direct and group messages. You can also choose specific content to filter.

> **Note**
>
> Amazon Kendra now supports an upgraded Slack connector.
> The console has been automatically upgraded for you. Any new connectors you create in the console will use the upgraded architecture. If you use the API, you must now use the TemplateConfiguration object instead of the `SlackConfiguration` object to configure your connector.
> Connectors configured using the older console and API architecture will continue to function as configured. However, you won't be able to edit or update them. If you want to edit or update your connector configuration, you must create a new connector.
> We recommended migrating your connector workflow to the upgraded version. Support for connectors configured using the older architecture is scheduled to end by June 2024.

You can connect Amazon Kendra to your Slack data source using the Amazon Kendra console or the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Slack data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more

Supported features

Amazon Kendra Slack data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters

- Full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Slack data source, make these changes in your Slack and AWS accounts.

In Slack, make sure you have:

- Configured a Slack Bot User OAuth token or Slack User OAuth token. You can choose either token to connect Amazon Kendra to your Slack data source. A token is required to use as your authentication credentials. See Slack documentation on access tokens for more information.

  > **Note**
  >
  > If you use the bot token as part of your Slack credentials, you cannot index direct messages and group messages and you must add the bot token to the channel you want to index.

  > **Note**
  >
  > We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

- Noted your Slack workspace team ID from your Slack workspace main page URL. For example, *`https://app.slack.com/client/T0123456789/...`* where *`T0123456789`* is the team ID.
- Added the following Oauth scopes/permissions:

User token scope	Bot token scope
• channels:history • channels:read • emoji:read	• channels:history • channels:manage • channels:read

User token scope	Bot token scope
• files:read • groups:history • groups:read • im:history • im:read • mpim:history • mpim:read • team:read • users.profile:read • users:read • users:read.email	• conversations.connect:manage • conversations.connect:read • files:read • groups:history • groups:read • im:history • im:read • mpim:history • mpim:read • reactions:read • team:read • usergroups:read • users.profile:read • users:read • users:read.email

- Checked each document is unique in Slack and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Slack authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Slack data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Slack data source, you must provide the necessary details of your Slack data source so that Amazon Kendra can access your data. If you have not yet configured Slack for Amazon Kendra, see [Prerequisites](#).

Console

To connect Amazon Kendra to Slack

1. Sign in to the AWS Management Console and open the [Amazon Kendra console](#).
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Slack connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Slack connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

   a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.
   b. (Optional) **Description**—Enter an optional description for your data source.
   c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.
   d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.
   e. Choose **Next**.
6. On the **Define access and security** page, enter the following information:
   a. For **Slack workspace team ID**—The team ID of your Slack workspace. You can find your team ID in your Slack workspace main page URL. For example, *https://app.slack.com/client/T0123456789/...* where *T0123456789* is the team ID.
   b. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see [User context filtering].
   c. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Slack authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.
      i. Enter following information in the **Create an AWS Secrets Manager secret window**:
         A. **Secret name**—A name for your secret. The prefix 'AmazonKendra-Slack-' is automatically added to your secret name.
         B. For **Slack token**—Enter the authentication credential values you configured Slack.
      ii. Save and add your secret.
   d. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.
   e. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to

filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

f. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

g. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

a. **Select type of content**—Select the Slack entities or content types you want to crawl. You can choose from all channels, public channels, private channels, group messages, and private messages.

b. **Select crawl start date**—Enter the date you want to start crawling your content.

c. For **Additional configuration**—Choose to include bot and archived messages and use regular expression patterns to include or exclude certain content.

> **Note**
>
> If you choose to include for both channel IDs and channel names, the Amazon Kendra Slack connector will prioritize channel IDs over channel names.
> If you've chosen to include certain private and group messages, the Amazon Kendra Slack connector will ignore all private and group messages and only crawl the private and group messages you specify.

d. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your

data if your initial sync failed, even if you don't choose full sync as your sync mode option.

- Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
- New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.

e. In **Sync run schedule**, for **Frequency**—Choose how often to sync your data source content and update your index.

f. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

a. **Default data source fields**—Select from the Amazon Kendra generated default data source fields you want to map to your index.

b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Slack

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `SLACK` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Slack workspace team ID**—The Slack team ID you copied from your Slack main page URL.

- **Since date**—The date to start crawling your data from your Slack workspace team. The date must follow this format: yyyy-mm-dd.
- **Sync mode**—Specify how Amazon Kendra should update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option. You can choose between:
  - `FORCED_FULL_CRAWL` to freshly index all content, replacing existing content each time your data source syncs with your index.
  - `FULL_CRAWL` to index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
  - `CHANGE_LOG` to index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
- **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.
- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Slack account. The secret is stored in a JSON structure with the following keys:

```
{
    "slackToken": "token"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Slack connector and Amazon Kendra. For more information, see IAM roles for Slack data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see [Configuring Amazon Kendra to use an Amazon VPC](#).
- **Specific channels**—Filter by public or private channels, and specify certain channels by their ID.
- **Types of channels and messages**—Whether Amazon Kendra should index your public and private channels, your group and direct messages, and your bot and archived messages. If you use a bot token as part of your Slack authentication credentials, you must add the bot token to the channel you want to index. You cannot index direct messages and group messages using a bot token.
- **Look back**—You can choose to configure a `lookBack` parameter so that the Slack connector crawls updated or deleted content up to a specified number of hours before your last connector sync.
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain Slack content. If you use a bot token as part of your Slack authentication credentials, you must add the bot token to the channel you want to index. You cannot index direct messages and group messages using a bot token.

  > **Note**
  >
  > Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **Field mappings**—Choose to map your Slack data source fields to your Amazon Kendra index fields. For more information, see [Mapping data source fields](#).

  > **Note**
  >
  > The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must

map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Slack template schema.

Learn more

To learn more about integrating Amazon Kendra with your Slack data source, see:

- Unravel the knowledge in Slack workspaces with intelligent search using the Amazon Kendra Slack connector

Zendesk

Zendesk is a customer relationship management system that helps businesses automate and enhance customer support interactions. You can use Amazon Kendra to index your Zendesk support tickets, ticket comments, ticket attachments, help center articles, article comments, article comment attachments, guide community topics, community posts, and community post comments.

You can filter by organization name if you want to index tickets that are only within a specific organization. You can also choose to set a crawl date for when you want to start crawling data from Zendesk.

You can connect Amazon Kendra to your Zendesk data source using the Amazon Kendra console and the TemplateConfiguration API.

For troubleshooting your Amazon Kendra Zendesk data source connector, see Troubleshooting data sources.

Topics

- Supported features
- Prerequisites
- Connection instructions
- Learn more
- Notes

Supported features

Amazon Kendra Zendesk data source connector supports the following features:

- Field mappings
- User access control
- Inclusion/exclusion filters
- Change log, full and incremental content syncs
- Virtual private cloud (VPC)

Prerequisites

Before you can use Amazon Kendra to index your Zendesk data source, make these changes in your Zendesk and AWS accounts.

In Zendesk, make sure you have:

- Created a Zendesk Suite (Professional/Enterprise) administrative account.
- Noted your Zendesk host URL. For example, *`https://{sub-domain}.zendesk.com/`*.

> **Note**
>
> (On-premise/server) Amazon Kendra checks if the endpoint information included in AWS Secrets Manager is the same the endpoint information specified in your data source configuration details. This helps protect against the confused deputy problem, which is a security issue where a user doesn't have permission to perform an action but uses Amazon Kendra as a proxy to access the configured secret and perform the action. If you later change your endpoint information, you must create a new secret to sync this information.

- Set up OAuth 2.0 Authentication :
  1. In Admin Center, navigate to Apps and integrations > APIs > Zendesk API.
  2. Select the OAuth Clients tab and click "Add OAuth client".
  3. Configure the folllwing OAuth client details: Set Client Name and Description, Set Client Kind to "Confidential", Add appropriate Redirect URLs, Save and securely store the generated Client Secret.

4. Ensure the OAuth client has the required "read" scope (or "read write" if you need write access).
5. Generate an Access Token using the implicit grant flow:
   - In a browser, navigate to: `https://{subdomain}.zendesk.com/oauth/authorizations/new?response_type=token&client_id=<your_client_id>&scope=read`
   - Authenticate and authorize the application when prompted.
   - After authorization, you'll be redirected to a URL containing the access token in the fragment (after the #). Extract this token.
6. Store the generated access token securely. This is the "Implicit grant token" you'll use for Kendra integration.

- > **Note**
  >
  > We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).
- **Optional:** Installed an SSL certificate to allow Amazon Kendra to connect.
- Checked each document is unique in Zendesk and across other data sources you plan to use for the same index. Each data source that you want to use for an index must not contain the same document across the data sources. Document IDs are global to an index and must be unique per index.

In your AWS account, make sure you have:

- Created an Amazon Kendra index and, if using the API, noted the index ID.
- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

> **Note**
>
> If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.

- Stored your Zendesk authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

> **Note**
>
> We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We do **not** recommend that you re-use credentials and secrets across data sources, and connector versions 1.0 and 2.0 (where applicable).

If you don't have an existing IAM role or secret, you can use the console to create a new IAM role and Secrets Manager secret when you connect your Zendesk data source to Amazon Kendra. If you are using the API, you must provide the ARN of an existing IAM role and Secrets Manager secret, and an index ID.

Connection instructions

To connect Amazon Kendra to your Zendesk data source, you must provide the necessary details of your Zendesk data source so that Amazon Kendra can access your data. If you have not yet configured Zendesk for Amazon Kendra, see Prerequisites.

Console

To connect Amazon Kendra to Zendesk

1. Sign in to the AWS Management Console and open the Amazon Kendra console.
2. From the left navigation pane, choose **Indexes** and then choose the index you want to use from the list of indexes.

   > **Note**
   >
   > You can choose to configure or edit your **User access control** settings under **Index settings**.

3. On the **Getting started** page, choose **Add data source**.
4. On the **Add data source** page, choose **Zendesk connector**, and then choose **Add connector**. If using version 2 (if applicable), choose **Zendesk connector** with the "V2.0" tag.
5. On the **Specify data source details** page, enter the following information:

a. In **Name and description**, for **Data source name**—Enter a name for your data source. You can include hyphens but not spaces.

b. (Optional) **Description**—Enter an optional description for your data source.

c. In **Default language**—Choose a language to filter your documents for the index. Unless you specify otherwise, the language defaults to English. Language specified in the document metadata overrides the selected language.

d. In **Tags**, for **Add new tag**—Include optional tags to search and filter your resources or track your AWS costs.

e. Choose **Next**.

6. On the **Define access and security** page, enter the following information:

a. **Zendesk URL**—Enter your Zendesk URL. For example, *`https://{sub-domain}.zendesk.com/`*.

b. **Authorization**—Turn on or off access control list (ACL) information for your documents, if you have an ACL and want to use it for access control. The ACL specifies which documents that users and groups can access. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see [User context filtering](#).

c. **AWS Secrets Manager secret**—Choose an existing secret or create a new Secrets Manager secret to store your Zendesk authentication credentials. If you choose to create a new secret an AWS Secrets Manager secret window opens.

i. Create a new secret with the following structure:

```
{
         "hostUrl": "https://yoursubdomain.zendesk.com/",
         "accessToken": "your_implicit_grant_access_token"
}
```

ii. Save and add your secret.

d. **Virtual Private Cloud (VPC)**—You can choose to use a VPC. If so, you must add **Subnets** and **VPC security groups**.

e. **Identity crawler**—Specify whether to turn on Amazon Kendra's identity crawler. The identity crawler uses the access control list (ACL) information for your documents to filter search results based on the user or their group access to documents. If you have an ACL for your documents and choose to use your ACL, you can then also choose to turn on Amazon Kendra's identity crawler to configure user context filtering of search results. Otherwise, if identity crawler is turned off, all documents can be publicly searched. If you want to use access control for your documents and identity crawler is turned off, you can alternatively use the PutPrincipalMapping API to upload user and group access information for user context filtering.

f. **IAM role**—Choose an existing IAM role or create a new IAM role to access your repository credentials and index content.

> **Note**
>
> IAM roles used for indexes cannot be used for data sources. If you are unsure if an existing role is used for an index or FAQ, choose **Create a new role** to avoid errors.

g. Choose **Next**.

7. On the **Configure sync settings** page, enter the following information:

   a. **Select contents**—Select the types of content you want to crawl from tickets, to help center articles, community topics, and more.

   b. **Organization name**—Enter the Zendesk organization names to filter content.

   c. **Sync start date**—Enter the date from which you want to start crawling your content.

   d. **Regex patterns**—Add regular expression patterns to include or exclude certain files. You can add up to 100 patterns.

   e. **Sync mode**—Choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Kendra for the first time, all content is crawled and indexed by default. You must run a full sync of your data if your initial sync failed, even if you don't choose full sync as your sync mode option.

   - Full sync: Freshly index all content, replacing existing content each time your data source syncs with your index.
   - New, modified sync: Index only new and modified content each time your data source syncs with your index. Amazon Kendra can use your data source's mechanism for tracking content changes and index content that changed since the last sync.
   - New, modified, deleted sync: Index only new, modified, and deleted content each time your data source syncs with your index. Amazon Kendra can use your data

source's mechanism for tracking content changes and index content that changed since the last sync.

f. In **Sync run schedule** for **Frequency**—Choose how often to sync your data source content and update your index.

g. Choose **Next**.

8. On the **Set field mappings** page, enter the following information:

    a. **Default data source fields**—Select from the Amazon Kendra generated default data source fields you want to map to your index.

    b. **Add field**—To add custom data source fields to create an index field name to map to and the field data type.

    c. Choose **Next**.

9. On the **Review and create** page, check that the information you have entered is correct and then select **Add data source**. You can also choose to edit your information from this page. Your data source will appear on the **Data sources** page after the data source has been added successfully.

API

To connect Amazon Kendra to Zendesk

You must specify a JSON of the data source schema using the TemplateConfiguration API. You must provide the following information:

- **Data source**—Specify the data source type as `ZENDESK` when you use the TemplateConfiguration JSON schema. Also specify the data source as `TEMPLATE` when you call the CreateDataSource API.
- **Host URL**—Provide your Zendesk host URL as part of the connection configuration or repository endpoint details. For example, *`https://yoursubdomain.zendesk.com`*.
- **Change log**—Whether Amazon Kendra should use the Zendesk data source change log mechanism to determine if a document must be updated in the index.

> **Note**
>
> Use the change log if you don't want Amazon Kendra to scan all of the documents. If your change log is large, it might take Amazon Kendra less time to scan the

documents in the Zendesk data source than to process the change log. If you are syncing your Zendesk data source with your index for the first time, all documents are scanned.

- **Secret Amazon Resource Name (ARN)**—Provide the Amazon Resource Name (ARN) of an Secrets Manager secret that contains the authentication credentials for your Zendesk account. The secret is stored in a JSON structure with the following keys:

```
{
    "hostUrl": "https://yoursubdomain.zendesk.com",
    "clientId": "client ID",
    "clientSecret": "Zendesk client secret",
    "userName": "Zendesk user name",
    "password": "Zendesk password"
}
```

- **IAM role**—Specify `RoleArn` when you call `CreateDataSource` to provide an IAM role with permissions to access your Secrets Manager secret and to call the required public APIs for the Zendesk connector and Amazon Kendra. For more information, see IAM roles for Zendesk data sources.

You can also add the following optional features:

- **Virtual Private Cloud (VPC)**—Specify `VpcConfiguration` when you call `CreateDataSource`. For more information, see Configuring Amazon Kendra to use an Amazon VPC.
- **Document/content types**—Specify whether to crawl:
  - Support tickets, ticket comments, and/or ticket comment attachments
  - Help center articles, article attachments, and article comments
  - Guide community topics, posts, or post comments
- **Inclusion and exclusion filters**—Specify whether to include or exclude certain Slack content. If you use a bot token as part of your Slack authentication credentials, you must add the bot token to the channel you want to index. You cannot index direct messages and group messages using a bot token.

> **Note**
>
> Most data sources use regular expression patterns, which are inclusion or exclusion patterns referred to as filters. If you specify an inclusion filter, only content that matches the inclusion filter is indexed. Any document that doesn't match the inclusion filter isn't indexed. If you specify an inclusion and exclusion filter, documents that match the exclusion filter are not indexed, even if they match the inclusion filter.

- **User context filtering and access control**—Amazon Kendra crawls the access control list (ACL) for your documents, if you have an ACL for your documents. The ACL information is used to filter search results based on the user or their group access to documents. For more information, see User context filtering.
- **Field mappings**—Choose to map your Zendesk data source fields to your Amazon Kendra index fields. For more information, see Mapping data source fields.

> **Note**
>
> The document body field or the document body equivalent for your documents is required in order for Amazon Kendra to search your documents. You must map your document body field name in your data source to the index field name `_document_body`. All other fields are optional.

For a list of other important JSON keys to configure, see Zendesk template schema.

Learn more

To learn more about integrating Amazon Kendra with your Zendesk data source, see:

- Discover insights from Zendesk with Amazon Kendra intelligent search

Notes

- When Access Control Lists (ACLs) are enabled, the "Sync only new or modified content" option is not available due to Zendesk API limitations. We recommend using "Full sync" or "New, modified, or deleted content sync" modes instead, or disable ACLs if you need to use this sync mode.

Mapping data source fields

Amazon Kendra data source connectors can map document or content fields from your data source to fields in your Amazon Kendra index. By default, each connector is designed to crawl specific data source fields. Default data source fields and their properties cannot be changed or customized. On the Amazon Kendra console, default fields and default field properties that cannnot be edited are grayed out.

Amazon Kendra connectors also allow you to map custom document or content fields from your data source to custom fields in your index. For example, if you have a field in your data source called "dept" that contains department information for a document, you can map it to an index field called "Department". That way, you can use the field when querying documents.

You can also map Amazon Kendra reserved or common fields such as `_created_at`. If your data source has a field called "creation_date", you can map this to the equivalent Amazon Kendra reserved field called `_created_at`. For more information on Amazon Kendra reserved fields, see [Document attributes or fields](#).

You can map fields for most data sources. You can create field mappings for the following data sources:

- Adobe Experience Manager
- Alfresco
- Aurora (MySQL)
- Aurora (PostgreSQL)
- Amazon FSx (Windows)
- Amazon FSx (NetApp ONTAP)
- Amazon RDS/Aurora
- Amazon RDS (Microsoft SQL Server)
- Amazon RDS (MySQL)
- Amazon RDS (Oracle)
- Amazon RDS (PostgreSQL)
- Amazon Kendra Web Crawler
- WorkDocs
- Box

- Confluence
- Dropbox
- Drupal
- GitHub
- Google Workspace Drives
- Gmail
- IBM DB2
- Jira
- Microsoft Exchange
- Microsoft OneDrive
- Microsoft SharePoint
- Microsoft Teams
- Microsoft SQL Server
- Microsoft Yammer
- MySQL
- Oracle Database
- PostgreSQL
- Quip
- Salesforce
- ServiceNow
- Slack
- Zendesk

If you store your documents in an S3 bucket, or S3 data source, you specify your fields using a JSON metadata file. For more information, see S3 data source connector.

Mapping your data source fields to an index field is a three-step process:

1. Create an index. For more information, see Creating an index.
2. Update the index to add fields.
3. Create a data source and include field mappings to map reserved fields and any custom fields to Amazon Kendra index fields.

To update the index to add custom fields, use the console to edit the data source field mappings and add a custom field or use the UpdateIndex API. You can add a total of 500 custom fields to your index.

For database data sources, if the name of the database column matches the name of a reserved field, the field and column are automatically mapped.

With the UpdateIndex API, you add reserved and custom fields using `DocumentMetadataConfigurationUpdates`.

The following JSON example uses `DocumentMetadataConfigurationUpdates` to add a field called "Department" to the index.

```
"DocumentmetadataConfigurationUpdates": [
   {
       "Name": "Department",
       "Type": "STRING_VALUE"
   }
]
```

When you create the field, you have the option of setting how the field is used for search. You can choose from the following:

- **Displayable**—Determines whether the field is returned in the query response. The default is `true`.
- **Facetable**—Indicates that the field can be used to create facets. The default is `false`.
- **Searchable**—Determines whether the field is used in the search. The default is `true` for string fields and `false` for number and date fields.
- **Sortable**—Indicates that the field can be used to sort the response from a query. Can only be set for date, number, and string fields. Can't be set for string list fields.

The following JSON example uses `DocumentMetadataConfigurationUpdates` to add a field called "Department" to the index and marks it as facetable.

```
"DocumentMetadataConfigurationUpdates": [
   {
       "Name": "Department",
       "Type": "STRING_VALUE",
       "Search": {
```

```
            "Facetable": true
        }
    }
]
```

Using Amazon Kendra reserved or common document fields

With the UpdateIndex API, you can create reserved or common fields using `DocumentMetadataConfigurationUpdates` and specifying the Amazon Kendra reserved index field name to map to your equivalent document attribute/field name. You can also create custom fields. If you use a data source connector, most include field mappings that map your data source document fields to Amazon Kendra index fields. If you use the console, you update fields by selecting your data source, selecting the edit action, and then proceeding next to the field mappings section for configuring the data source.

You can configure the `Search` object to set a field as either displayable, facetable, searchable, and sortable. You can configure the `Relevance` object to set a field's rank order, boost duration or time period to apply to boosting, freshness, importance value, and importance values mapped to specific field values. If you use the console, you can set the search settings for a field by selecting the facet option in the navigation menu. To set relevance tuning, select the option to search your index in the navigation menu, enter a query, and use the side panel options to tune the search relevance. You cannot change the field type once you have created the field.

Amazon Kendra has the following reserved or common document fields that you can use:

- `_authors`—A list of one or more authors responsible for the content of the document.
- `_category`—A category that places a document in a specific group.
- `_created_at`—The date and time in ISO 8601 format that the document was created. For example, 2012-03-25T12:30:10+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.
- `_data_source_id`—The identifier of the data source that contains the document.
- `_document_body`—The content of the document.
- `_document_id`—A unique identifier for the document.
- `_document_title`—The title of the document.
- `_excerpt_page_number`—The page number in a PDF file where the document excerpt appears. If your index was created before September 8, 2020, you must re-index your documents before you can use this attribute.

- `_faq_id`—If this is a question-answer type document (FAQ), a unique identifier for the FAQ.
- `_file_type`—The file type of the document, such as pdf or doc.
- `_last_updated_at`—The date and time in ISO 8601 format that the document was last updated. For example, 2012-03-25T12:30:10+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.
- `_source_uri`—The URI where the document is available. For example, the URI of the document on a company website.
- `_version`—An identifier for the specific version of a document.
- `_view_count`—The number of times that the document has been viewed.
- `_language_code` (String)—The code for a language that applies to the document. This defaults to English if you do not specify a language. For more information on supported languages, including their codes, see Adding documents in languages other than English.

For custom fields, you create these fields using `DocumentMetadataConfigurationUpdates` with the `UpdateIndex` API, just as you do when creating a reserved or common field. You must set the appropriate data type for your custom field. If you use the console, you update fields by selecting your data source, selecting the edit action, and then proceeding next to the field mappings section for configuring the data source. Some data sources don't support adding new fields or custom fields. You cannot change the field type once you have created the field.

The following are the types you can set for custom fields:

- Date
- Number
- String
- String list

If you added documents to the index using BatchPutDocument API, `Attributes` lists the fields/attributes of your documents and you create fields using the `DocumentAttribute` object.

For documents indexed from an Amazon S3 data source, you create fields using a JSON metadata file that includes the fields information.

If you use a supported database as your data source, you can configure your fields using the field mappings option.

Adding documents in languages other than English

You can index documents in multiple languages. If you don't specify a language, Amazon Kendra indexes documents in English by default. You include the language code for a document in the document metadata as a field. See Field mappings and Custom attributes for more information on the `_language_code` field for a document.

You can specify the language code for all your documents in your data source when you call CreateDataSource. If a document doesn't have a language code specified in a metadata field, the document is indexed using the language code specified for all documents at the data source level. In the console, you can index documents in a supported language only at the data source level. Go to **Data sources**, then the **Specify data source details** page, and choose a language from the dropdown **Language**.

You can also search or query documents in a supported language. For more information, see Searching in languages.

The following languages and their codes are supported (English or `en` is supported by default if you don't specify a language). This table includes languages that Amazon Kendra supports with full semantic search, as well as languages that only support simple keyword matching. Languages that support full semantic search are marked with an asterisk and are in bold text in the following table. English (default language) is also supported with full semantic search.

Language name	Language code
Arabic	`ar`
Armenian	`hy`
Basque	`eu`
Bengali	`bn`
Bulgarian	`bg`
Catalan	`ca`
Chinese – simplified and traditional*	`zh`
Czech	`cs`

Language name	Language code
Danish	`da`
Dutch	`nl`
Finnish	`fi`
French – includes French (Canada)*	`fr`
Galician	`gl`
German*	`de`
Greek	`el`
Hindi	`hi`
Hungarian	`hu`
Indonesian	`id`
Irish	`ga`
Italian	`it`
Japanese*	`ja`
Korean*	`ko`
Latvian	`lv`
Lithuanian	`lt`
Norwegian	`no`
Persian	`fa`
Portuguese	`pt`
Portuguese (Brazil)*	`pt-BR`

Language name	Language code
Romanian	ro
Russian	ru
Sorani	ckb
Spanish – includes Spanish (Mexico)*	es
Swedish	sv
Turkish	tr

**Semantic search is supported for the language.*

For languages that support semantic search, the following features are supported.

- Document relevance beyond simple keyword matching.
- FAQs beyond simple keyword matching.
- Extracting answers from documents based on Amazon Kendra's reading comprehension.
- Confidence buckets (very high, high, medium, and low) of the search results.

For languages that don't support semantic search, simple keyword matching is supported for document relevance and FAQs.

Synonyms (including custom synonyms), incremental learning and feedback, and query suggestions are only supported for English (default language).

Configuring Amazon Kendra to use an Amazon VPC

Amazon Kendra can connect to a virtual private cloud (VPC) that you created with Amazon Virtual Private Cloud to index content stored in data sources running in your private cloud. When you create a data source connector, you can provide security group and subnet identifiers for the subnet that contains your data source. With this information, Amazon Kendra creates an elastic network interface that it uses to securely communicate with your data source within your VPC.

To set up an Amazon Kendra data source connector with Amazon VPC, you can use either the AWS Management Console or the CreateDataSource API operation. If you use the console, you connect a VPC during the connector configuration process.

> **Note**
>
> The Amazon VPC feature is optional when setting up an Amazon Kendra data source connector. If your data source is accessible from the public internet, you don't need to enable the Amazon VPC feature. Not all Amazon Kendra data source connectors support Amazon VPC.

If your data source isn't running on Amazon VPC and isn't accessible from the public internet, you first connect your data source to your VPC using a virtual private network (VPN). Then, you can connect your data source to Amazon Kendra by using a combination of Amazon VPC and AWS Virtual Private Network. For information about setting up a VPN, see the AWS VPN documentation.

Topics

- Configuring Amazon VPC support for Amazon Kendra connectors
- Set up an Amazon Kendra data source to connect to Amazon VPC
- Connecting to a database in a VPC
- Troubleshooting VPC connection issues

Configuring Amazon VPC support for Amazon Kendra connectors

To configure Amazon VPC for use with your Amazon Kendra connectors, take the following steps.

Steps

- Step 1. Create Amazon VPC subnets for Amazon Kendra
- Step 2. Create Amazon VPC security groups for Amazon Kendra
- Step 3. Configure your external data source and Amazon VPC

Step 1. Create Amazon VPC subnets for Amazon Kendra

Create or choose an existing Amazon VPC subnet that Amazon Kendra can use to access your data source. The prepared subnets must be in one of the following AWS Regions and Availability Zones:

- US West (Oregon)/us-west-2—usw2-az1, usw2-az2, usw2-az3
- US East (N. Virginia)/us-east-1—use1-az1, use1-az2, use1-az4
- US East (Ohio)/us-east-2—use2-az1, use2-az2, use2-az3
- Asia Pacific (Tokyo)/ap-northeast-1—apne1-az1, apne1-az2, apne1-az4
- Asia Pacific (Mumbai)/ap-south-1—aps1-az1, aps1-az2, aps1-az3
- Asia Pacific (Singapore)/ap-southeast-1—apse1-az1, apse1-az2, apse1-az3
- Asia Pacific (Sydney)/ap-southeast-2—apse2-az1, apse2-az2, apse2-az3
- Canada (Central)/ca-central-1—cac1-az1, cac1-az2, cac1-az4
- Europe (Ireland)/eu-west-1—euw1-az1, uew1-az2, euw1-az3
- Europe (London)/eu-west-2—euw2-az1, euw2-az2, euw2-az3

Your data source must be accessible from the subnets that you provided to Amazon Kendra connector.

For more information about how to configure Amazon VPC subnets, see Subnets for your Amazon VPC in the *Amazon VPC User Guide*.

If Amazon Kendra must route the connection between two or more subnets, you can prepare multiple subnets. For example, the subnet that contains your data source is out of IP addresses. In that case, you can provide Amazon Kendra with an additional subnet that has sufficient IP addresses and connected to the first subnet. If you list multiple subnets, the subnets must be able to communicate with each other.

Step 2. Create Amazon VPC security groups for Amazon Kendra

To connect your Amazon Kendra data source connector to Amazon VPC, you must prepare one or more security groups from your VPC to assign to Amazon Kendra. The security groups will be associated to the elastic network interface created by Amazon Kendra. This network interface controls inbound and outbound traffic to and from Amazon Kendra when accessing the Amazon VPC subnets.

Make sure that your security group's outbound rules allow the traffic from Amazon Kendra data source connectors to access the subnets and the data source that you are going to sync with. For example, you might use an MySQL connector to sync from a MySQL database. If you're using the default port, the security groups must allow Amazon Kendra to access port 3306 on the host that runs the database.

We recommend that you configure a default security group with the following values for Amazon Kendra to use:

- **Inbound rules** – If you choose to leave this empty, all inbound traffic will be blocked.
- **Outbound rules** – Add one rule to allow all outbound traffic so that Amazon Kendra can initiate the requests to sync from your data source.
  - **IP version** – IPv4
  - **Type** – All traffic
  - **Protocol** – All traffic
  - **Port range** – All
  - **Destination** – 0.0.0.0/0

For more information about how to configure Amazon VPC security groups, see Security group rules in the *Amazon VPC User Guide*.

Step 3. Configure your external data source and Amazon VPC

Make sure that your external data source has the correct permissions configuration and network settings for Amazon Kendra to access it. You can find detailed instructions on how to configure your data sources in the prerequisites section of each connector page.

Also, check your Amazon VPC settings and make sure that your external data source is reachable from the subnet you will assign to Amazon Kendra. To do this, we recommend that you create an Amazon EC2 instance in the same subnet with the same security groups and test access to your data source from this Amazon EC2 instance. For more information, see Troubleshooting Amazon VPC connection.

Set up an Amazon Kendra data source to connect to Amazon VPC

When you add a new data source in Amazon Kendra, you can use the Amazon VPC feature if the selected data source connector supports this feature.

You can set up a new Amazon Kendra data source with Amazon VPC enabled by using the AWS Management Console or the Amazon Kendra API. Specifically, use the CreateDataSource API operation, and then use the `VpcConfiguration` parameter to provide the following information:

- `SubnetIds` – A list of identifiers of Amazon VPC subnets
- `SecurityGroupIds` – A list of identifiers of Amazon VPC security groups

If you use the console, you provide the required Amazon VPC information during connector configuration. To use the console to enable the Amazon VPC feature for a connector, you first choose an Amazon VPC. Then, you provide identifiers of any Amazon VPC subnets and identifiers of any Amazon VPC security groups. You can choose the Amazon VPC subnets and Amazon VPC security groups that you created in Configuring Amazon VPC, or use any existing ones.

Topics

- Viewing Amazon VPC identifiers
- Checking your data source IAM role

Viewing Amazon VPC identifiers

The identifiers for subnets and security groups are configured in the Amazon VPC console. To view the identifiers, use the following procedures.

To view subnet identifiers

1. Sign in to the AWS Management Console and open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. From the navigation pane, choose **Subnets**.
3. From the **Subnets** list, choose the subnet that contains your database server.
4. From the **Details** tab, make a note of the identifier in the **Subnet ID** field.

To view security group identifiers

1. Sign in to the AWS Management Console and open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. From the navigation pane, choose **Security groups**.
3. From the security group list, choose the group that you want the identifier for.
4. From the **Details** tab, make a note of the identifier in the **Security Group ID** field.

Checking your data source IAM role

Make sure that your data source connector AWS Identity and Access Management IAM) role contains permissions to access your Amazon VPC.

If you use the console to create a new role for your IAM role, Amazon Kendra automatically adds the correct permissions to your IAM role on your behalf. If you use the API, or use an existing IAM role, check that your role contains permissions to access Amazon VPC. To verify that you have the right permissions, see IAM roles for VPC.

You can modify an existing data source to use a different Amazon VPC subnet. However, check your data source's IAM role and, if necessary, modify it to reflect the change for the Amazon Kendra data source connector to work properly.

Connecting to a database in a VPC

The following example shows how to connect a MySQL database running in a virtual private cloud (VPC) . The example assumes that you're starting with your default VPC and that you need to create a MySQL database. If you already have a VPC, make sure that it's configured as shown. If you have a MySQL database, you can use that instead of creating a new one.

Steps

- Step 1: Configure a VPC
- Step 2: Create and configure security groups
- Step 3: Create a database
- Step 4: Create a data source connector

Step 1: Configure a VPC

Configure your VPC so that you have a private subnet and a security group for Amazon Kendra to access a MySQL database running in the subnet. The subnets provided in the VPC configuration must be in the US West (Oregon) Region, the US East (N. Virginia) Region, or the Europe (Ireland) Region.

To configure a VPC using Amazon VPC

1. Sign in to the AWS Management Console and open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. From the navigation pane, choose **Route tables**, then choose **Create route table**.
3. For the **Name** field, enter **`Private subnet route table`**. From the **VPC** dropdown, select your VPC, and then choose **Create route table**. Choose **Close** to return to the list of route tables.

4. From the navigation pane, choose **NAT gateways**, then choose **Create NAT gateway**.
5. From the **Subnet** dropdown, choose the subnet that's the public subnet. Make a note of the subnet ID.
6. If you don't have an Elastic IP address, choose **Create New EIP**, choose **Create a NAT Gateway**, and then choose **Close**.
7. From the navigation pane, choose **Route tables**.
8. From the route table list, choose the **Private subnet route table** that you created in step 3. From **Actions**, choose **Edit routes**.
9. Choose **Add route**. For the destination, enter **`0.0.0.0/0`** to allow all outgoing traffic to the internet. For **Target**, choose **NAT Gateway**, and then choose the gateway that you created in step 4. Choose **Save changes**, and then choose **Close**.
10. From **Actions**, choose **Edit subnet associations**.
11. Choose the subnets that you want to be private. Don't choose the subnet with the NAT gateway that you noted previously. Choose **Save associations** when you're done.

Step 2: Create and configure security groups

Next, configure security groups for your database.

To create and configure security groups

1. Sign in to the AWS Management Console and open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. From the description of your VPC, note the IPv4 CIDR.
3. From the navigation pane, choose **Security groups** and then choose **Create security group**.
4. For **Security group name**, enter **`DataSourceInboundSecurityGroup`**. Provide a description, then choose your VPC from the list. Choose **Create security group** and then choose **Close**.
5. Choose the **Inbound rules** tab.
6. Choose **Edit inbound rules**, and then choose **Add rule**
7. For a database, enter the port number for the **Port range**. For example, for MySQL it's **`3306`**, and, for HTTPS, it's **`443`**. For the **Source**, type the Classless Inter-Domain Routing (CIDR) of your VPC. Choose **Save rules** and then choose **Close**.

The security group allows anyone within the VPC to connect to the database, and it allows outbound connections to the internet.

Step 3: Create a database

Create a database to hold your documents, or you can use your existing database.

For instructions on how to create a MySQL database, see MySQL.

Step 4: Create a data source connector

After you configure your VPC and create your database, you can create a data source connector for the database. For information about database connectors that Amazon Kendra supports, see Supported connectors.

For your database, make sure that you configure your VPC, the private subnets that you created in your VPC, and the security group that you created in your VPC.

Troubleshooting VPC connection issues

If you encounter any issues with your virtual private cloud (VPC) connection, check that your IAM permissions, security group settings, and the subnet's route tables are configured correctly.

One potential cause of a failed data source connector sync is that the data source might be unreachable from the subnet that you assigned to Amazon Kendra. To troubleshoot this issue, we recommend that you create an Amazon EC2 instance with the same Amazon VPC settings. Then, try to access the data source from this Amazon EC2 instance using REST API calls or other methods (based on the specific type of your data source).

If you successfully access the data source from the Amazon EC2 instance that you create, it means your data source is reachable from this subnet. Therefore, your sync issue isn't related to your data source being inaccessible by Amazon VPC.

If you can't access your Amazon EC2 instance from your VPC configuration and validate it with the Amazon EC2 instance that you created, you need to troubleshoot further. For example, if you have an Amazon S3 connector whose sync failed with errors about connection issues, you can set up an Amazon EC2 instance with the same Amazon VPC configuration that you assigned to your Amazon S3 connector. Then, use this Amazon EC2 instance to test if your Amazon VPC has been set up correctly.

The following is an example of setting up an Amazon EC2 instance to troubleshoot your Amazon VPC connection with an Amazon S3 data source.

Topics

Step 1: Launch an Amazon EC2 instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Select **Launch an instance**.
3. Choose **Network settings**, and then choose **Edit**, and then do the following:

    a. Choose the same VPC and **Subnet** that you assigned to Amazon Kendra.

    b. For **Firewall (security groups)**, choose **Select existing security group**. Then, select the security group that you assigned to Amazon Kendra.

    > **Note**
    >
    > The security group should allow outbound traffic to Amazon S3.

    c. Set **Auto-assign public IP** to **Disable**.

    d. In **Advanced details**, do the following:

    - For **IAM instance profile**, select **Create new IAM profile** to create and attach an IAM instance profile to your instance. Make sure that the profile has permissions to access Amazon S3. For more information, see How can I grant my Amazon EC2 instance access to an Amazon S3 bucket? in AWS re:Post.
    - Leave all other settings as default.

    e. Review and launch the Amazon EC2 instance.

Step 2: Connect to Amazon EC2 instance

After your Amazon EC2 instance is running, go to your instance detail page and connect to your instance. To do so, use the steps in [Connect to your instances without requiring a public IPv4 address using EC2 Instance Connect Endpoint] in the *Amazon EC2 User Guide for Linux Instances*.

Step 3: Test Amazon S3 access

After you have connected to your Amazon EC2 instance terminal, run an AWS CLI command to test the connection from this private subnet to your Amazon S3 bucket.

To test Amazon S3 access, type the following AWS CLI command in the AWS CLI: `aws s3 ls`

After the AWS CLI command runs, review the following:

- If you've set up the necessary IAM permissions correctly and your Amazon S3 setup is correct, you should see a list of your Amazon S3 buckets.
- If you see permission errors such as `Access Denied`, it's likely that yourVPC configuration is correct, but something is wrong with your IAM permissions or Amazon S3 bucket policy.

If the command is timing out, then it's likely that your connection is timing out because your VPC setup is incorrect and the Amazon EC2 instance can't access Amazon S3 from your subnet. Reconfigure your VPC, and try again.

Deleting an index, data source, or batch uploaded documents

This section shows you how to delete an index, a data source respository of documents in your index, or documents in your index that you batch uploaded.

Topics

Deleting an index

You can delete an index from Amazon Kendra when you are no longer using the index. For example, delete an index when:

- You are no longer using the index and want to reduce charges to your AWS account. An Amazon Kendra index accrues charges while it is running whether or not you make queries on the index.
- You want to reconfigure the index for a different edition of Amazon Kendra. Delete the existing index and then create a new one with the different edition.
- You have reached the maximum number of indexes in your account and don't want to exceed your quota. Delete an existing index and add a new one. For information about the maximum number of indexes that you can create, see Quotas.

To delete an index, use the console, the AWS Command Line Interface, the AWS CloudFormation script, or the `DeleteIndex` API. Deleting an index removes the index and all associated data sources and document data. Deleting an index doesn't remove the original documents from your storage.

Deleting an index is an asynchronous operation. When you start deleting an index, the index status changes to `DELETING`. It remains in the `DELETING` state until all of the information related to the index is removed. Once the index is deleted, it no longer appears in the results of a call to the ListIndices API. If you call the DescribeIndex API with the deleted index's identifier, you receive and `ResourceNotFound` exception.

To delete an index (console)

1. Sign in to the AWS Management Console and open the Amazon Kendra console at https://console.aws.amazon.com/kendra/.
2. In the navigation pane, choose **Indexes**, and then choose the index to delete.
3. Choose **Delete** to delete the selected index.

To delete an index (CLI)

- In the AWS CLI, use the following command. The command is formatted for Linux and macOS. If you are using Windows, replace the Unix line continuation character (\) with a caret (^).

```
aws kendra delete-index \
   --id index-id
```

Deleting a data source

You delete a data source when you want to remove the information contained in the data source from your Amazon Kendra index. For example, delete a data source when:

- A data source is incorrectly configured. Delete the data source, wait for the data source to finish deleting, and then recreate it.
- You migrated documents from one data source to another. Delete the original data source and recreate it in the new location.
- You have reached the limit of data sources for an index. Delete one of the existing data sources and add a new one. For more information about the number of data sources that you can create, see Quotas.

To delete a data source, use the console, the AWS Command Line Interface (AWS CLI), the `DeleteDataSource` API, or a AWS CloudFormation script. Deleting a data source removes all of the information about the data source from the index. If you only want to stop synching the data source, change the synchronization schedule for the data source to "run on demand".

Deleting a data source is an asynchronous operation. When you start deleting a data source, the data source status changes to `DELETING`. It remains in the `DELETING` state until the information related to the data source is removed. After the data source is deleted, it no longer appears in the results of a call to the ListDataSources API. If you call the DescribeDataSource API with the deleted data source's identifier, you receive a `ResourceNotFound` exception.

> **Note**
>
> Deleting an entire data source or re-syncing your index after deleting specific documents from a data source could take up to an hour or more, depending on the number of documents you want to delete.

To delete a data source (console)

1. Sign in to the AWS Management Console and open the Amazon Kendra console at https://console.aws.amazon.com/kendra/.
2. In the navigation pane, choose **Indexes**, and then choose the index that contains the data source to delete.
3. In the navigation pane, choose **Data sources**.
4. Choose the data source to remove.
5. Choose **Delete** to delete the data source.

To delete a data source (CLI)

- In the AWS Command Line Interface, use the following command. The command is formatted for Linux and macOS. If you are using Windows, replace the Unix line continuation character (\) with a caret (^).

  ```
  aws kendra delete-data-source \
     --id data-source-id \
     --index-id index-id
  ```

When you delete a data source, Amazon Kendra removes all of the stored information about the data source. Amazon Kendra removes all of the document data stored in the index, and all run histories and metrics associated with the data source. Deleting a data source does not remove the original documents from your storage.

Documents in the data source may be included in the document count returned by the `DescribeIndex` API while Amazon Kendra deletes a data source. Documents from the data source may appear in search results while Amazon Kendra deletes the data source.

Amazon Kendra releases the resources for a data source as soon as you call the `DeleteDataSource` API or choose to delete the data source in the console. If you are deleting the data source to reduce the number of data sources below your limit, you can create a new data source right away.

If you are deleting a data source and then creating another data source to the document data, wait for the first data source to be deleted before you sync the new data source.

You can delete a data source that is in the process of syncing with Amazon Kendra. The sync is stopped and the data source is removed. If you attempt to start a sync when the data source is being deleted, you get a `ConflictException` exception.

You can't delete a data source if the associated index is in the `DELETING` state. Deleting an index deletes all of the data sources for the index. You can start deleting an index while a data source for that index is in the `DELETING` state.

If you have two data sources pointing to the same documents, such as two data sources pointing to the same Amazon S3 bucket, documents in the index might be inconsistent when one of the data sources is deleted. When two data sources reference the same documents, only one copy of the document data is stored in the index. Removing one data source removes the index data for the documents. The other data source is not aware that the documents have been removed, so Amazon Kendra won't correctly re-index the documents the next time it syncs. When you have two data sources pointing to the same document location, you should delete both data sources and then recreate one.

Deleting batch uploaded documents

You can delete documents directly from an index using the BatchDeleteDocument API. You can't delete documents directly using the console. If you use the console, you can either delete specific documents from your data source repository and re-sync with your index or delete the entire data source connector.

Deleting documents from an index using `BatchDeleteDocument` is an asynchronous operation. After you call the `BatchDeleteDocument` API, you use the BatchGetDocumentStatus API to monitor the progress of deleting your documents. When a document is deleted from the index, Amazon Kendra returns NOT_FOUND as the status.

> **Note**
>
> Deleting documents from an index using BatchDeleteDocument could take up to an hour or more, depending on the number of documents you want to delete.

To delete batch uploaded documents from an index (CLI)

- In the AWS Command Line Interface, use the following command. The command is formatted for Linux and macOS. If you are using Windows, replace the Unix line continuation character (\) with a caret (^).

```
aws kendra batch-delete-document \
   --index-id index-id \
   --document-id-list 'doc-id-1' 'doc-id-2'
```

Enriching your documents during ingestion

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

You can alter your content and document metadata fields or attributes during the document ingestion process. With Amazon Kendra's *Custom Document Enrichment* feature, you can create, modify, or delete document attributes and content when you ingest your documents into Amazon Kendra. This means you can manipulate and ingest your data as you need.

This feature gives you control over how your documents are treated and ingested into Amazon Kendra. For example, you can scrub personally identifiable information in the document metadata while ingesting your documents into Amazon Kendra.

Another way that you can use this feature is to invoke a Lambda function in AWS Lambda to run Optical Character Recognition (OCR) on images, translation on text, and other tasks for preparing the data for search or analysis. For example, you can invoke a function to run OCR on images. The function could interpret text from images and treat each image as a textual document. A company that receives mailed-in customer surveys and stores these surveys as images could ingest these images as textual documents into Amazon Kendra. The company can then search for valuable customer survey information in Amazon Kendra.

You can use basic operations to apply as a first parse of your data, and then use a Lambda function to apply more complex operations on your data. For example, you could use a basic operation to simply remove all values in the document metadata field 'Customer_ID', and then apply a Lambda function to extract text from images of the text in the documents.

How Custom Document Enrichment works

The overall process of Custom Document Enrichment is as follows:

1. You configure Custom Document Enrichment when you create or update your data source, or index your documents directly into Amazon Kendra.
2. Amazon Kendra applies inline configurations or basic logic to alter your data. For more information, see the section called "Basic operations to change metadata".

3. If you choose to configure advanced data manipulation, Amazon Kendra can apply this on your original, raw documents or on the structured, parsed documents. For more information, see the section called "Lambda functions: extract and change metadata or content".
4. Your altered documents are ingested into Amazon Kendra.

At any point in this process, if your configuration is not valid, Amazon Kendra throws an error.

When you call CreateDataSource, UpdateDataSource, or BatchPutDocument APIs, you provide your Custom Document Enrichment configuration. If you call `BatchPutDocument`, you must configure Custom Document Enrichment with each request. If you use the console, you select your index and then select **Document enrichments** to configure Custom Document Enrichment.

If you use **Document enrichments** in the console, you can choose to only configure basic operations or only Lambda functions or both, like you can using the API. You can select **Next** in the console steps to choose not to configure basic operations and only Lambda functions, including whether to apply to the original (pre-extraction) or structured (post-extraction) data. You can only save your configurations by completing all the steps in the console. Your document configurations are not saved if you don't complete all the steps.

Basic operations to change metadata

You can manipulate your document fields and content using basic logic. This includes removing values in a field, modifying values in a field using a condition, or creating a field. For advanced manipulations that go beyond what you can manipulate using basic logic, invoke a Lambda function. For more information, see the section called "Lambda functions: extract and change metadata or content".

To apply basic logic, you specify the target field you want to manipulate using the DocumentAttributeTarget object. You provide the attribute key. For example, the key 'Department' is a field or attribute that holds all the department names associated with the documents. You can also specify a value to use in the target field if a certain condition is met. You set the condition using the DocumentAttributeCondition object. For example, if the 'Source_URI' field contains 'financial' in its URI value, then prefill the target field 'Department' with the target value 'Finance' for the document. You can also delete the values of the target document attribute.

To apply basic logic using the console, select your index and then select **Document enrichments** in the navigation menu. Go to **Configure basic operations** to apply basic manipulations to your document fields and content.

The following is an example of using basic logic to remove all customer identification numbers in the document field called 'Customer_ID'.

Example 1: Removing customer identification numbers associated with the documents

Data before basic manipulation applied.

Document_ID	Body_Text	Customer_ID
1	Lorem Ipsum.	CID1234
2	Lorem Ipsum.	CID1235
3	Lorem Ipsum.	CID1236

Data after basic manipulation applied.

Document_ID	Body_Text	Customer_ID
1	Lorem Ipsum.	
2	Lorem Ipsum.	
3	Lorem Ipsum.	

The following is an example of using basic logic to create a field called 'Department' and prefill this field with the department names based on information from the 'Source_URI' field. This uses the condition that if the 'Source_URI' field contains 'financial' in its URI value, then prefill the target field 'Department' with the target value 'Finance' for the document.

Example 2: Creating 'Department' field and prefilling it with department names associated with the documents using a condition.

Data before basic manipulation applied.

Document_ID	Body_Text	Source_URI
1	Lorem Ipsum.	financial/1

Document_ID	Body_Text	Source_URI
2	Lorem Ipsum.	financial/2
3	Lorem Ipsum.	financial/3

Data after basic manipulation applied.

Document_ID	Body_Text	Source_URI	Department
1	Lorem Ipsum.	financial/1	Finance
2	Lorem Ipsum.	financial/2	Finance
3	Lorem Ipsum.	financial/3	Finance

Note

Amazon Kendra can't create a target document field if it isn't already created as an index field. After you create your index field, you can create a document field using `DocumentAttributeTarget`. Amazon Kendra then maps your newly created document metadata field to your index field.

The following code is an example of configuring basic data manipulation to remove customer identification numbers associated with the documents.

Console

To configure basic data manipulation to remove customer identification numbers

1. In the left navigation pane, under **Indexes**, select **Document enrichments** and then select **Add document enrichment**.
2. On the **Configure basic operations** page, choose from the dropdown your data source that you want to alter document fields and content. Then choose from the dropdown the document field name 'Customer_ID', select from the dropdown the index field name

'Customer_ID', and select from the dropdown the target action **Delete**. Then select **Add basic operation**.

CLI

To configure basic data manipulation to remove customer identification numbers

```
aws kendra create-data-source \
 --name data-source-name \
 --index-id index-id \
 --role-arn arn:aws:iam::account-id:role/role-name \
 --type S3 \
 --configuration '{"S3Configuration":{"BucketName":"S3-bucket-name"}}' \
 --custom-document-enrichment-configuration '{"InlineConfigurations":[{"Target":
{"TargetDocumentAttributeKey":"Customer_ID", "TargetDocumentAttributeValueDeletion":
 true}}]}'
```

Python

To configure basic data manipulation to remove customer identification numbers

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Create a data source with customizations")

# Provide the name of the data source
name = "data-source-name"
# Provide the index ID for the data source
index_id = "index-id"
# Provide the IAM role ARN required for data sources
role_arn = "arn:aws:iam::${account-id}:role/${role-name}"
# Provide the data source connection information
data_source_type = "S3"
S3_bucket_name = "S3-bucket-name"
# Configure the data source with Custom Document Enrichment
configuration = {"S3Configuration":
        {
            "BucketName": S3_bucket_name
```

```
        }
    }
custom_document_enrichment_configuration = {"InlineConfigurations":[
        {
            "Target":{"TargetDocumentAttributeKey":"Customer_ID",
                       "TargetDocumentAttributeValueDeletion": True}
        }]
    }

try:
    data_source_response = kendra.create_data_source(
        Name = name,
        IndexId = index_id,
        RoleArn = role_arn,
        Type = data_source_type
        Configuration = configuration
        CustomDocumentEnrichmentConfiguration =
 custom_document_enrichment_configuration
    )

    pprint.pprint(data_source_response)

    data_source_id = data_source_response["Id"]

    print("Wait for Amazon Kendra to create the data source with your
 customizations.")

    while True:
        # Get the details of the data source, such as the status
        data_source_description = kendra.describe_data_source(
            Id = data_source_id,
            IndexId = index_id
        )
        status = data_source_description["Status"]
        print(" Creating data source. Status: "+status)
        time.sleep(60)
        if status != "CREATING":
            break

    print("Synchronize the data source.")

    sync_response = kendra.start_data_source_sync_job(
        Id = data_source_id,
        IndexId = index_id
```

```
    )

    pprint.pprint(sync_response)

    print("Wait for the data source to sync with the index.")

    while True:

        jobs = kendra.list_data_source_sync_jobs(
            Id= data_source_id,
            IndexId= index_id
        )

        # For this example, there should be one job
        status = jobs["History"][0]["Status"]

        print(" Syncing data source. Status: "+status)
        time.sleep(60)
        if status != "SYNCING":
            break

except  ClientError as e:
        print("%s" % e)

print("Program ends.")
```

Java

To configure basic data manipulation to remove customer identification numbers

```
package com.amazonaws.kendra;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.CreateDataSourceRequest;
import software.amazon.awssdk.services.kendra.model.CreateDataSourceResponse;
import software.amazon.awssdk.services.kendra.model.CreateIndexRequest;
import software.amazon.awssdk.services.kendra.model.CreateIndexResponse;
import software.amazon.awssdk.services.kendra.model.DataSourceConfiguration;
import software.amazon.awssdk.services.kendra.model.DataSourceStatus;
import software.amazon.awssdk.services.kendra.model.DataSourceSyncJob;
import software.amazon.awssdk.services.kendra.model.DataSourceSyncJobStatus;
import software.amazon.awssdk.services.kendra.model.DataSourceType;
import software.amazon.awssdk.services.kendra.model.DescribeDataSourceRequest;
```

```
import software.amazon.awssdk.services.kendra.model.DescribeDataSourceResponse;
import software.amazon.awssdk.services.kendra.model.DescribeIndexRequest;
import software.amazon.awssdk.services.kendra.model.DescribeIndexResponse;
import software.amazon.awssdk.services.kendra.model.IndexStatus;
import software.amazon.awssdk.services.kendra.model.ListDataSourceSyncJobsRequest;
import software.amazon.awssdk.services.kendra.model.ListDataSourceSyncJobsResponse;
import software.amazon.awssdk.services.kendra.model.S3DataSourceConfiguration;
import software.amazon.awssdk.services.kendra.model.StartDataSourceSyncJobRequest;
import software.amazon.awssdk.services.kendra.model.StartDataSourceSyncJobResponse;

public class CreateDataSourceWithCustomizationsExample {

    public static void main(String[] args) throws InterruptedException {
        System.out.println("Create a data source with customizations");

        String dataSourceName = "data-source-name";
        String indexId = "index-id";
        String dataSourceRoleArn = "arn:aws:iam::account-id:role/role-name";
        String s3BucketName = "S3-bucket-name"

        KendraClient kendra = KendraClient.builder().build();

        CreateDataSourceRequest createDataSourceRequest = CreateDataSourceRequest
            .builder()
            .name(dataSourceName)
            .description(experienceDescription)
            .roleArn(experienceRoleArn)
            .type(DataSourceType.S3)
            .configuration(
                DataSourceConfiguration
                    .builder()
                    .s3Configuration(
                        S3DataSourceConfiguration
                            .builder()
                            .bucketName(s3BucketName)
                            .build()
                    ).build()
            )
            .customDocumentEnrichmentConfiguration(
                CustomDocumentEnrichmentConfiguration
                    .builder()
                    .inlineConfigurations(Arrays.asList(
                        InlineCustomDocumentEnrichmentConfiguration
                            .builder()
```

```
                            .target(
                                DocumentAttributeTarget
                                    .builder()
                                    .targetDocumentAttributeKey("Customer_ID")
                                    .targetDocumentAttributeValueDeletion(true)
                                    .build())
                            .build()
                    )).build();

        CreateDataSourceResponse createDataSourceResponse =
kendra.createDataSource(createDataSourceRequest);
        System.out.println(String.format("Response of creating data source: %s",
createDataSourceResponse));

        String dataSourceId = createDataSourceResponse.id();
        System.out.println(String.format("Waiting for Kendra to create the data
source %s", dataSourceId));
        DescribeDataSourceRequest describeDataSourceRequest =
DescribeDataSourceRequest
            .builder()
            .indexId(indexId)
            .id(dataSourceId)
            .build();

        while (true) {
            DescribeDataSourceResponse describeDataSourceResponse =
kendra.describeDataSource(describeDataSourceRequest);

            DataSourceStatus status = describeDataSourceResponse.status();
            System.out.println(String.format("Creating data source. Status: %s",
status));
            TimeUnit.SECONDS.sleep(60);
            if (status != DataSourceStatus.CREATING) {
                break;
            }
        }

        System.out.println(String.format("Synchronize the data source %s",
dataSourceId));
        StartDataSourceSyncJobRequest startDataSourceSyncJobRequest =
StartDataSourceSyncJobRequest
            .builder()
            .indexId(indexId)
            .id(dataSourceId)
```

```
            .build();
        StartDataSourceSyncJobResponse startDataSourceSyncJobResponse =
 kendra.startDataSourceSyncJob(startDataSourceSyncJobRequest);
        System.out.println(String.format("Waiting for the data
 source to sync with the index %s for execution ID %s", indexId,
 startDataSourceSyncJobResponse.executionId()));

        // For this example, there should be one job
        ListDataSourceSyncJobsRequest listDataSourceSyncJobsRequest =
 ListDataSourceSyncJobsRequest
            .builder()
            .indexId(indexId)
            .id(dataSourceId)
            .build();

        while (true) {
            ListDataSourceSyncJobsResponse listDataSourceSyncJobsResponse =
 kendra.listDataSourceSyncJobs(listDataSourceSyncJobsRequest);
            DataSourceSyncJob job = listDataSourceSyncJobsResponse.history().get(0);
            System.out.println(String.format("Syncing data source. Status: %s",
 job.status()));

            TimeUnit.SECONDS.sleep(60);
            if (job.status() != DataSourceSyncJobStatus.SYNCING) {
                break;
            }

        }

        System.out.println("Data source creation with customizations is complete");
    }
}
```

Lambda functions: extract and change metadata or content

You can manipulate your document fields and content using Lambda functions. This is useful if you want to go beyond basic logic and apply advanced data manipulations. For example, using Optical Character Recognition (OCR), which interprets text from images, and treats each image as a textual document. Or, retrieving the current date-time in a certain time zone and inserting the date-time where there's an empty value for a date field.

You can apply basic logic first and then use a Lambda function to further manipulate your data, or vice versa. You can also choose to only apply a Lambda function.

Amazon Kendra can invoke a Lambda function to apply advanced data manipulations during the ingestion process as part of your CustomDocumentEnrichmentConfiguration. You specify a role that includes permission to execute the Lambda function and access your Amazon S3 bucket to store the output of your data manipulations—see IAM access roles.

Amazon Kendra can apply a Lambda function on your original, raw documents or on the structured, parsed documents. You can configure a Lambda function that takes your original or raw data and applies your data manipulations using PreExtractionHookConfiguration. You can also configure a Lambda function that takes your structured documents and applies your data manipulations using PostExtractionHookConfiguration. Amazon Kendra extracts the document metadata and text to structure your documents. Your Lambda functions must follow the mandatory request and response structures. For more information, see the section called “Data contracts for Lambda functions”.

To configure a Lambda function in the console, select your index and then select **Document enrichments** in the navigation menu. Go to **Configure Lambda functions** to configure a Lambda function.

You can configure only one Lambda function for `PreExtractionHookConfiguration` and and only one Lambda function for `PostExtractionHookConfiguration`. However, your Lambda function can invoke other functions that it requires. You can configure both `PreExtractionHookConfiguration` and `PostExtractionHookConfiguration` or either one. Your Lambda function for `PreExtractionHookConfiguration` must not exceed a run time of 5 minutes and your Lambda function for `PostExtractionHookConfiguration` must not exceed a run time of 1 minute. Configuring Custom Document Enrichment naturally takes longer to ingest your documents into Amazon Kendra than if you were to not configure this.

You can configure Amazon Kendra to invoke a Lambda function only if a condition is met. For example, you can specify a condition that if there are empty date-time values, then Amazon Kendra should invoke a function that inserts the current date-time.

The following is an example of using a Lambda function to run OCR to interpret text from images and store this text in a field called 'Document_Image_Text'.

Example 1: Extracting text from images to create textual documents

Data before advanced manipulation applied.

Document_ID	Document_Image
1	image_1.png
2	image_2.png
3	image_3.png

Data after advanced manipulation applied.

Document_ID	Document_Image	Document_Image_Text
1	image_1.png	Mailed survey response
2	image_2.png	Mailed survey response
3	image_3.png	Mailed survey response

The following is an example of using a Lambda function to insert the current date-time for empty date values. This uses the condition that if a date field value is 'null', then replace this with the current date-time.

Example 2: Replacing empty values in the Last_Updated field with the current date-time.

Data before advanced manipulation applied.

Document_ID	Body_Text	Last_Updated
1	Lorem Ipsum.	January 1, 2020
2	Lorem Ipsum.	
3	Lorem Ipsum.	July 1, 2020

Data after advanced manipulation applied.

Document_ID	Body_Text	Last_Updated
1	Lorem Ipsum.	January 1, 2020
2	Lorem Ipsum.	December 1, 2021
3	Lorem Ipsum.	July 1, 2020

The following code is an example of configuring a Lambda function for advanced data manipulation on the raw, original data.

Console

To configure a Lambda function for advanced data manipulation on the raw, original data

1. In the left navigation pane, under **Indexes**, select **Document enrichments** and then select **Add document enrichment**.
2. On the **Configure Lambda functions** page, in the **Lambda for pre-extraction** section, select from the dropdowns your Lambda function ARN and your Amazon S3 bucket. Add your IAM access role by selecting the option to create a new role from the dropdown. This creates the required Amazon Kendra permissions to create the document enrichment.

CLI

To configure a Lambda function for advanced data manipulation on the raw, original data

```
aws kendra create-data-source \
 --name data-source-name \
 --index-id index-id \
 --role-arn arn:aws:iam::account-id:role/role-name \
 --type S3 \
 --configuration '{"S3Configuration":{"BucketName":"S3-bucket-name"}}' \
 --custom-document-enrichment-configuration '{"PreExtractionHookConfiguration":
{"LambdaArn":"arn:aws:iam::account-id:function/function-name", "S3Bucket":"S3-
bucket-name"}, "RoleArn": "arn:aws:iam:account-id:role/cde-role-name"}'
```

Python

To configure a Lambda function for advanced data manipulation on the raw, original data

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Create a data source with customizations.")

# Provide the name of the data source
name = "data-source-name"
# Provide the index ID for the data source
index_id = "index-id"
# Provide the IAM role ARN required for data sources
role_arn = "arn:aws:iam::${account-id}:role/${role-name}"
# Provide the data source connection information
data_source_type = "S3"
S3_bucket_name = "S3-bucket-name"
# Configure the data source with Custom Document Enrichment
configuration = {"S3Configuration":
        {
            "BucketName": S3_bucket_name
        }
    }
custom_document_enrichment_configuration = {"PreExtractionHookConfiguration":
        {
            "LambdaArn":"arn:aws:iam::account-id:function/function-name",
            "S3Bucket":"S3-bucket-name"
        }
    "RoleArn":"arn:aws:iam::account-id:role/cde-role-name"
    }

try:
    data_source_response = kendra.create_data_source(
        Name = name,
        IndexId = index_id,
        RoleArn = role_arn,
        Type = data_source_type
        Configuration = configuration
        CustomDocumentEnrichmentConfiguration =
 custom_document_enrichment_configuration
    )
```

```
    pprint.pprint(data_source_response)

    data_source_id = data_source_response["Id"]

    print("Wait for Amazon Kendra to create the data source with your
 customizations.")

    while True:
        # Get the details of the data source, such as the status
        data_source_description = kendra.describe_data_source(
            Id = data_source_id,
            IndexId = index_id
        )
        status = data_source_description["Status"]
        print(" Creating data source. Status: "+status)
        time.sleep(60)
        if status != "CREATING":
            break

    print("Synchronize the data source.")

    sync_response = kendra.start_data_source_sync_job(
        Id = data_source_id,
        IndexId = index_id
    )

    pprint.pprint(sync_response)

    print("Wait for the data source to sync with the index.")

    while True:

        jobs = kendra.list_data_source_sync_jobs(
            Id = data_source_id,
            IndexId = index_id
        )

        # For this example, there should be one job
        status = jobs["History"][0]["Status"]

        print(" Syncing data source. Status: "+status)
        time.sleep(60)
        if status != "SYNCING":
            break
```

```
except  ClientError as e:
        print("%s" % e)

print("Program ends.")
```

Java

To configure a Lambda function for advanced data manipulation on the raw, original data

```
package com.amazonaws.kendra;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.CreateDataSourceRequest;
import software.amazon.awssdk.services.kendra.model.CreateDataSourceResponse;
import software.amazon.awssdk.services.kendra.model.CreateIndexRequest;
import software.amazon.awssdk.services.kendra.model.CreateIndexResponse;
import software.amazon.awssdk.services.kendra.model.DataSourceConfiguration;
import software.amazon.awssdk.services.kendra.model.DataSourceStatus;
import software.amazon.awssdk.services.kendra.model.DataSourceSyncJob;
import software.amazon.awssdk.services.kendra.model.DataSourceSyncJobStatus;
import software.amazon.awssdk.services.kendra.model.DataSourceType;
import software.amazon.awssdk.services.kendra.model.DescribeDataSourceRequest;
import software.amazon.awssdk.services.kendra.model.DescribeDataSourceResponse;
import software.amazon.awssdk.services.kendra.model.DescribeIndexRequest;
import software.amazon.awssdk.services.kendra.model.DescribeIndexResponse;
import software.amazon.awssdk.services.kendra.model.IndexStatus;
import software.amazon.awssdk.services.kendra.model.ListDataSourceSyncJobsRequest;
import software.amazon.awssdk.services.kendra.model.ListDataSourceSyncJobsResponse;
import software.amazon.awssdk.services.kendra.model.S3DataSourceConfiguration;
import software.amazon.awssdk.services.kendra.model.StartDataSourceSyncJobRequest;
import software.amazon.awssdk.services.kendra.model.StartDataSourceSyncJobResponse;


public class CreateDataSourceWithCustomizationsExample {

    public static void main(String[] args) throws InterruptedException {
        System.out.println("Create a data source with customizations");

        String dataSourceName = "data-source-name";
        String indexId = "index-id";
        String dataSourceRoleArn = "arn:aws:iam::account-id:role/role-name";
        String s3BucketName = "S3-bucket-name"
```

```
        KendraClient kendra = KendraClient.builder().build();

        CreateDataSourceRequest createDataSourceRequest = CreateDataSourceRequest
            .builder()
            .name(dataSourceName)
            .description(experienceDescription)
            .roleArn(experienceRoleArn)
            .type(DataSourceType.S3)
            .configuration(
                DataSourceConfiguration
                    .builder()
                    .s3Configuration(
                        S3DataSourceConfiguration
                            .builder()
                            .bucketName(s3BucketName)
                            .build()
                    ).build()
            )
            .customDocumentEnrichmentConfiguration(
                CustomDocumentEnrichmentConfiguration
                    .builder()
                    .preExtractionHookConfiguration(
                        HookConfiguration
                            .builder()
                            .lambdaArn("arn:aws:iam::account-id:function/function-
name")
                            .s3Bucket("S3-bucket-name")
                            .build())
                    .roleArn("arn:aws:iam::account-id:role/cde-role-name")
                    .build();

        CreateDataSourceResponse createDataSourceResponse =
 kendra.createDataSource(createDataSourceRequest);
        System.out.println(String.format("Response of creating data source: %s",
 createDataSourceResponse));

        String dataSourceId = createDataSourceResponse.id();
        System.out.println(String.format("Waiting for Kendra to create the data
 source %s", dataSourceId));
        DescribeDataSourceRequest describeDataSourceRequest =
 DescribeDataSourceRequest
            .builder()
            .indexId(indexId)
```

```
            .id(dataSourceId)
            .build();

        while (true) {
            DescribeDataSourceResponse describeDataSourceResponse =
kendra.describeDataSource(describeDataSourceRequest);

            DataSourceStatus status = describeDataSourceResponse.status();
            System.out.println(String.format("Creating data source. Status: %s",
status));
            TimeUnit.SECONDS.sleep(60);
            if (status != DataSourceStatus.CREATING) {
                break;
            }
        }

        System.out.println(String.format("Synchronize the data source %s",
dataSourceId));
        StartDataSourceSyncJobRequest startDataSourceSyncJobRequest =
StartDataSourceSyncJobRequest
            .builder()
            .indexId(indexId)
            .id(dataSourceId)
            .build();
        StartDataSourceSyncJobResponse startDataSourceSyncJobResponse =
kendra.startDataSourceSyncJob(startDataSourceSyncJobRequest);
        System.out.println(String.format("Waiting for the data
source to sync with the index %s for execution ID %s", indexId,
startDataSourceSyncJobResponse.executionId()));

        // For this example, there should be one job
        ListDataSourceSyncJobsRequest listDataSourceSyncJobsRequest =
ListDataSourceSyncJobsRequest
            .builder()
            .indexId(indexId)
            .id(dataSourceId)
            .build();

        while (true) {
            ListDataSourceSyncJobsResponse listDataSourceSyncJobsResponse =
kendra.listDataSourceSyncJobs(listDataSourceSyncJobsRequest);
            DataSourceSyncJob job = listDataSourceSyncJobsResponse.history().get(0);
            System.out.println(String.format("Syncing data source. Status: %s",
job.status()));
```

```
                TimeUnit.SECONDS.sleep(60);
                if (job.status() != DataSourceSyncJobStatus.SYNCING) {
                    break;
                }

            }

            System.out.println("Data source creation with customizations is complete");
        }
    }
```

Data contracts for Lambda functions

Your Lambda functions for advanced data manipulation interact with Amazon Kendra data contracts. The contracts are the mandatory request and response structures of your Lambda functions. If your Lambda functions don't follow these structures, then Amazon Kendra throws an error.

Your Lambda function for `PreExtractionHookConfiguration` should expect the following request structure:

```
{
    "version": <str>,
    "dataBlobStringEncodedInBase64": <str>, //In the case of a data blob
    "s3Bucket": <str>, //In the case of an S3 bucket
    "s3ObjectKey": <str>, //In the case of an S3 bucket
    "metadata": <Metadata>
}
```

The `metadata` structure, which includes the `CustomDocumentAttribute` structure, is as follows:

```
{
    "attributes": [<CustomDocumentAttribute<]
}

CustomDocumentAttribute
{
    "name": <str>,
    "value": <CustomDocumentAttributeValue>
```

```
}

CustomDocumentAttributeValue
{
    "stringValue": <str>,
    "integerValue": <int>,
    "longValue": <long>,
    "stringListValue": list<str>,
    "dateValue": <str>
}
```

Your Lambda function for `PreExtractionHookConfiguration` must adhere to the following response structure:

```
{
    "version": <str>,
    "dataBlobStringEncodedInBase64": <str>, //In the case of a data blob
    "s3ObjectKey": <str>, //In the case of an S3 bucket
    "metadataUpdates": [<CustomDocumentAttribute>]
}
```

Your Lambda function for `PostExtractionHookConfiguration` should expect the following request structure:

```
{
    "version": <str>,
    "s3Bucket": <str>,
    "s3ObjectKey": <str>,
    "metadata": <Metadata>
}
```

Your Lambda function for `PostExtractionHookConfiguration` must adhere to the following response structure:

```
PostExtractionHookConfiguration Lambda Response
{
    "version": <str>,
    "s3ObjectKey": <str>,
    "metadataUpdates": [<CustomDocumentAttribute>]
}
```

Your altered document is uploaded to your Amazon S3 bucket. The altered document must follow the format shown in the section called "Structured document format".

Structured document format

Amazon Kendra uploads your structured document to the given Amazon S3 bucket. The structured document follows this format:

```
Kendra document

{
   "textContent": <TextContent>
}

TextContent
{
  "documentBodyText": <str>
}
```

Example of a Lambda function that adheres to data contracts

The following Python code is an example of a Lambda function that applies advanced manipulation of the metadata fields `_authors`, `_document_title`, and the body content on the raw or original documents.

In the case of the body content residing in an Amazon S3 bucket

```
import json
import boto3

s3 = boto3.client("s3")

# Lambda function for advanced data manipulation
def lambda_handler(event, context):
    # Get the value of "S3Bucket" key name or item from the given event input
    s3_bucket = event.get("s3Bucket")
    # Get the value of "S3ObjectKey" key name or item from the given event input
    s3_object_key = event.get("s3ObjectKey")

    content_object_before_CDE = s3.get_object(Bucket = s3_bucket, Key = s3_object_key)
    content_before_CDE = content_object_before_CDE["Body"].read().decode("utf-8");
```

```
    content_after_CDE = "CDEInvolved " + content_before_CDE

    # Get the value of "metadata" key name or item from the given event input
    metadata = event.get("metadata")
    # Get the document "attributes" from the metadata
    document_attributes = metadata.get("attributes")

    s3.put_object(Bucket = s3_bucket, Key = "dummy_updated_kendra_document",
 Body=json.dumps(content_after_CDE))
    return {
        "version": "v0",
        "s3ObjectKey": "dummy_updated_kendra_document",
        "metadataUpdates": [
            {"name":"_document_title", "value":
{"stringValue":"title_from_pre_extraction_lambda"}},
            {"name":"_authors", "value":{"stringListValue":["author1", "author2"]}}
        ]
    }
```

In the case of the body content residing in a data blob

```
import json
import boto3
import base64

# Lambda function for advanced data manipulation
def lambda_handler(event, context):

    # Get the value of "dataBlobStringEncodedInBase64" key name or item from the given
 event input
    data_blob_string_encoded_in_base64 = event.get("dataBlobStringEncodedInBase64")
    # Decode the data blob string in UTF-8
    data_blob_string =
 base64.b64decode(data_blob_string_encoded_in_base64).decode("utf-8")
    # Get the value of "metadata" key name or item from the given event input
    metadata = event.get("metadata")
    # Get the document "attributes" from the metadata
    document_attributes = metadata.get("attributes")

    new_data_blob = "This should be the modified data in the document by pre processing
 lambda ".encode("utf-8")
    return {
        "version": "v0",
```

```
        "dataBlobStringEncodedInBase64":
 base64.b64encode(new_data_blob).decode("utf-8"),
        "metadataUpdates": [
            {"name":"_document_title", "value":
{"stringValue":"title_from_pre_extraction_lambda"}},
            {"name":"_authors", "value":{"stringListValue":["author1", "author2"]}}
        ]
    }
```

The following Python code is an example of a Lambda function that applies advanced manipulation of the metadata fields `_authors`, `_document_title`, and the body content on the structured or parsed documents.

```
import json
import boto3
import time

s3 = boto3.client("s3")

# Lambda function for advanced data manipulation
def lambda_handler(event, context):

    # Get the value of "S3Bucket" key name or item from the given event input
    s3_bucket = event.get("s3Bucket")
    # Get the value of "S3ObjectKey" key name or item from the given event input
    s3_key = event.get("s3ObjectKey")
    # Get the value of "metadata" key name or item from the given event input
    metadata = event.get("metadata")
    # Get the document "attributes" from the metadata
    document_attributes = metadata.get("attributes")

    kendra_document_object = s3.get_object(Bucket = s3_bucket, Key = s3_key)
    kendra_document_string = kendra_document_object['Body'].read().decode('utf-8')
    kendra_document = json.loads(kendra_document_string)
    kendra_document["textContent"]["documentBodyText"] = "Changing document body to a
 short sentence."

    s3.put_object(Bucket = s3_bucket, Key = "dummy_updated_kendra_document",
 Body=json.dumps(kendra_document))

    return {
        "version" : "v0",
        "s3ObjectKey": "dummy_updated_kendra_document",
```

```
        "metadataUpdates": [
            {"name": "_document_title", "value":{"stringValue":
"title_from_post_extraction_lambda"}},
            {"name": "_authors", "value":{"stringListValue":["author1", "author2"]}}
        ]
    }
```

Searching an index

Note

Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

To search an Amazon Kendra index, you use the Query API. The `Query` API returns information about the indexed documents that you use in your application. This section shows you how to make a query, perform filters, and interpret the response that you get from the `Query` API.

To search documents that you have indexed with Amazon Kendra for Amazon Lex, use AMAZON.KendraSearchIntent. For an example of configuring Amazon Kendra with Amazon Lex, see Creating a FAQ Bot for an Amazon Kendra Index.

Topics

Querying an index

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

When you search your index, Amazon Kendra uses all the information that you provided about your documents to determine the documents most relevant to the search terms entered. Some of the items that Amazon Kendra considers are:

- The text or body of the document.
- The title of the document.
- Custom text fields that you have marked as searchable.
- The date field that you have indicated should be used to determine the "freshness" of a document.
- Any other field that could provide relevant information.

Amazon Kendra can also filter the response based on any field/attribute filters that you might have set for the search. For example, if you have a custom field called "department", you can filter the response to return only documents from a department called "legal". For more information, see Custom fields or attributes.

Returned search results are sorted by the relevance that Amazon Kendra determines for each document. The results are paginated so that you can show a page at a time to your user.

To search documents that you have indexed with Amazon Kendra for Amazon Lex, use AMAZON.KendraSearchIntent. For an example of configuring Amazon Kendra with Amazon Lex, see Creating a FAQ Bot for an Amazon Kendra Index.

The following example shows how to search an index. Amazon Kendra determines the type of the search result (answer, document, question-answer) that's best suited for the query. You can't configure Amazon Kendra to return a specific type of search response (answer, document, question-answer) to a query.

For information about the query responses, see Query responses and response types.

Topics

Prerequisites

Before using the Query API to query an index:

- Set up the required permissions for an index and connect to your data source or batch upload your documents. For more information, see IAM roles. You use the Amazon Resource Name of the role when you call the API to create an index and data source connector or batch upload of documents.
- Set up either the AWS Command Line Interface, an SDK, or go to the Amazon Kendra console. For more information, see Setting up Amazon Kendra.
- Create an index and connect to a data source of documents or batch upload documents. For more information, see Creating an index and Creating a data source connector.

Searching an index (console)

You can use the Amazon Kendra console to search and test your index. You can make queries and see the results.

To search an index with the console

1. Sign in to the AWS Management Console and open the Amazon Kendra console at http://console.aws.amazon.com/kendra/.
2. On the navigation pane, choose **Indexes**.
3. Choose your index.
4. In the navigation menu, choose the option to search your index.
5. Enter a query in the text box and then press enter.

6. Amazon Kendra returns the results of the search.

You can also get the query ID for the search by selecting the lightbulb icon in the side panel.

Searching an index (SDK)

To search an index with Python or Java

- The following example searches an index. Change the value of `query` to your search query and `index_id` or `indexId` to the index identifier of the index that you want to search.

  You can also get the query ID for the search as part of the response elements when you call the [Query](#) API.

  Python

```
import boto3
import pprint

kendra = boto3.client("kendra")

# Provide the index ID
index_id = "index-id"
# Provide the query text
query = "query text"

response = kendra.query(
        QueryText = query,
        IndexId = index_id)

print("\nSearch results for query: " + query + "\n")

for query_result in response["ResultItems"]:

    print("-------------------")
    print("Type: " + str(query_result["Type"]))

    if query_result["Type"]=="ANSWER" or
 query_result["Type"]=="QUESTION_ANSWER":
        answer_text = query_result["DocumentExcerpt"]["Text"]
        print(answer_text)
```

```
    if query_result["Type"]=="DOCUMENT":
        if "DocumentTitle" in query_result:
            document_title = query_result["DocumentTitle"]["Text"]
            print("Title: " + document_title)
        document_text = query_result["DocumentExcerpt"]["Text"]
        print(document_text)

    print("------------------\n\n")
```

Java

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.QueryRequest;
import software.amazon.awssdk.services.kendra.model.QueryResponse;
import software.amazon.awssdk.services.kendra.model.QueryResultItem;

public class SearchIndexExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        String query = "query text";
        String indexId = "index-id";

        QueryRequest queryRequest = QueryRequest
            .builder()
            .queryText(query)
            .indexId(indexId)
            .build();

        QueryResponse queryResponse = kendra.query(queryRequest);

        System.out.println(String.format("\nSearch results for query: %s",
 query));
        for(QueryResultItem item: queryResponse.resultItems()) {
            System.out.println("----------------------");
            System.out.println(String.format("Type: %s", item.type()));

            switch(item.type()) {
                case QUESTION_ANSWER:
                case ANSWER:
```

```
                    String answerText = item.documentExcerpt().text();
                    System.out.println(answerText);
                    break;
                case DOCUMENT:
                    String documentTitle = item.documentTitle().text();
                    System.out.println(String.format("Title: %s",
 documentTitle));
                    String documentExcerpt = item.documentExcerpt().text();
                    System.out.println(String.format("Excerpt: %s",
 documentExcerpt));
                    break;
                default:
                    System.out.println(String.format("Unknown query result type:
 %s", item.type()));

            }

            System.out.println("-----------------------\n");
        }
    }
}
```

Searching an index (Postman)

You can use Postman to query and test your Amazon Kendra index.

To search an index using Postman

1. Create a new collection in Postman and set the request type to **POST**.
2. Enter the endpoint URL. For example, *https://kendra.<region>.amazonaws.com*.
3. Select the **Authorization** tab and enter the following information.
   - **Type**—Select **AWS signature**.
   - **AccessKey**—Enter the access key generated when you create an IAM user.
   - **SecretKey**—Enter the secret key generated when you create an IAM user.
   - **AWS Region**—Enter the region of you index. For example, *us-west-2*.
   - **Service Name**—Enter *kendra*. This is case sensitive, so must be lower case.

> **Warning**
>
> If you enter the incorrect service name or don't use lowercase, an error is thrown once you select **Send** to send the request: "Credential should be scoped to the correct service 'kendra'."
> You must also check that you entered the correct access key and secret key.

4. Select the **Headers** tab and enter the following key and value information.

   - Key: *X-Amz-Target*

     Value: *com.amazonaws.kendra.AWSKendraFrontendService.Query*
   - Key: *Content-Encoding*

     Value: *amz-1.0*
5. Select the **Body** tab and do the following.

   - Choose the **raw** JSON type for the body of the request.
   - Enter a JSON that includes your index ID and query text.

```
{
    "IndexId": "index-id",
    "QueryText": "enter a query here"
}
```

> **Warning**
>
> If your JSON doesn't use the correct indendation, an error is thrown: "SerializationException". Check the indendation in your JSON.

6. Select **Send** (near the top right).

Searching with advanced query syntax

You can create queries that are more specific than simple keyword or natural language queries by using advanced query syntax or operators. This includes ranges, Booleans, wildcards, and more. By using operators, you can give your query more context and further refine the search results.

Amazon Kendra supports the following operators.

- Boolean: Logic to limit or broaden the search. For example, `amazon AND sports` limits the search to only search for documents containing both terms.
- Parentheses: Reads nested query terms in order of precedence. For example, `(amazon AND sports) NOT rainforest` reads `(amazon AND sports)` before `NOT rainforest`.
- Ranges: Date or numeric range values. Ranges can be inclusive, exclusive, or unbounded. For example, you can search for documents that were last updated between January 1st 2020 and December 31st 2020, inclusive of these dates.
- Fields: Uses a specific field to limit the search. For example, you can search for documents that have 'United States' in the field 'location'.
- Wildcards: Partially match a string of text. For example, `Cloud*` could match CloudFormation. Amazon Kendra currently only supports trailing wildcards.
- Exact quotes: Exact match a string of text. For example, documents that contain `"Amazon Kendra" "pricing"`.

You can use a combination of any of the above operators.

Note that excessive use of operators or highly complex queries could impact query latency. Wildcards are some of the most expensive operators in terms of latency. A general rule is the more terms and operators that you use, the greater potential impact on latency. Other factors that affect latency include the average size of documents indexed, the size of your index, any filtering on search results, and the overall load on your Amazon Kendra index.

Boolean

You can combine or exclude words using the Boolean operators `AND`, `OR`, `NOT`.

The following are examples of using Boolean operators.

`amazon AND sports`

Returns search results that contain both the terms 'amazon' and 'sports' in the text, such as Amazon Prime video sports or other similar content.

`sports OR recreation`

Returns search results that contain the terms 'sports' or 'recreation', or both, in the text.

`amazon NOT rainforest`

Returns search results that contain the term 'amazon' but not the term 'rainforest' in the text. This is to search for documents about the company Amazon, not the Amazon Rainforest.

Parentheses

You can query nested words in order of precedence by using parentheses. The parentheses indicate to Amazon Kendra how a query should be read.

The following are examples of using parentheses operators.

`(amazon AND sports) NOT rainforest`

Returns documents that contain both the terms 'amazon' and 'sports' in the text, but not the term 'rainforest'. This is to search Amazon Prime video sports or other similar content, not adventure sports in the Amazon Rainforest. The parentheses help indicate that `amazon AND sports` should be read before `NOT rainforest`. The query should not be read as `amazon AND (sports NOT rainforest)`.

`(amazon AND (sports OR recreation)) NOT rainforest`

Returns documents that contain the terms 'sports' or 'recreation', or both, and the term 'amazon'. But it does not include the term 'rainforest'. This is to search Amazon Prime video sports or recreation, not adventure sports in the Amazon Rainforest. The parentheses help indicate that `sports OR recreation` should be read before combining with 'amazon', which is read before `NOT rainforest`. The query should not be read as `amazon AND (sports OR (recreation NOT rainforest))`.

Ranges

You can use a range of values to filter the search results. You specify an attribute and the range values. This can be date or numeric type.

Date ranges must be in the following formats:

- Epoch
- YYYY
- YYYY-mm
- YYYY-mm-dd
- YYYY-mm-dd'T'HH

You can also specify whether to include or exclude the lower and higher values of the range.

The following are examples of using range operators.

`_processed_date:>2019-12-31 AND _processed_date:<2021-01-01`

Returns documents that were processed in 2020—greater than December 31st 2019 and less than January 1st 2021.

`_processed_date:>=2020-01-01 AND _processed_date:<=2020-12-31`

Returns documents that were processed in 2020—greater than or equal to January 1st 2020 and less than or equal to December 31st 2020.

`_document_likes:<1`

Returns documents with zero likes or no user feedback—less than 1 like.

You can specify whether a range should be treated as inclusive or exclusive of the given range values.

Inclusive

`_last_updated_at:[2020-01-01 TO 2020-12-31]`

Returns documents last updated in 2020—includes the days December 1st 2020 and December 31st 2020.

Exclusive

`_last_updated_at:{2019-12-31 TO 2021-01-01}`

Returns documents last updated in 2020—excludes the days December 31st 2019 and January 1st 2021.

For unbounded ranges that are neither inclusive or exclusive, simply use the < and > operators. For example, `_last_updated_at:>2019-12-31 AND _last_updated_at:<2021-01-01`

Fields

You can limit your search to only return documents that meet a value in a specific field. The field can be of any type.

The following are examples of using field-level context operators.

`status:"Incomplete" AND financial_year:2021`

Returns documents for the 2021 financial year with their status as incomplete.

`(sports OR recreation) AND country:"United States" AND level:"professional"`

Returns documents that discuss professional sports or recreation in the United States.

Wildcards

You can broaden your search to account for variants of words and phrases using the wildcard operator. This is useful when searching for name variants. Amazon Kendra currently only supports trailing wildcards. The number of prefix characters for a trailing wildcard must be greater than two.

The following are examples of using wildcard operators.

`Cloud*`

Returns documents that contain variants such as CloudFormation and CloudWatch.

`kendra*aws`

Returns documents that contain variants such as kendra.amazonaws.

`kendra*aws*`

Returns documents that contain variants such as kendra.amazonaws.com

Exact quotes

You can use quotation marks to search for an exact match of a piece of text.

The following are examples of using quotation marks.

`"Amazon Kendra" "pricing"`

Returns documents that contain both the phrase 'Amazon Kendra' and the term 'pricing'. Documents must contain both 'Amazon Kendra' and 'pricing' in order to return in the results.

`"Amazon Kendra" "pricing" cost`

Returns documents that contain both the phrase 'Amazon Kendra' and the term 'pricing', and optionally the term 'cost'. Documents must contain both 'Amazon Kendra' and 'pricing' in order to return in the results, but might not necessarily include 'cost'.

Invalid query syntax

Amazon Kendra issues a warning if there are problems with your query syntax or your query is currently not supported by Amazon Kendra. For more information, see the API documentation for query warnings.

The following queries are examples of invalid query syntax.

`_last_updated_at:<2021-12-32`

Invalid date. Day 32 does not exist in the Gregorian calendar, which is used by Amazon Kendra.

`_view_count:ten`

Invalid numeric value. Digits must be used to represent numeric values.

`nonExistentField:123`

Invalid field search. The field must exist in order to use field search.

`Product:[A TO D]`

Invalid range. Numeric values or dates must be used for ranges.

`OR Hello`

Invalid Boolean. Operators must be used with terms and placed between terms.

Searching in languages

You can search for documents in a supported language. You pass the language code in the AttributeFilter to return filtered documents in your chosen language. You can type the query in a supported language.

If you do not specify a language, Amazon Kendra queries documents in English by default. For more information on supported languages, including their codes, see Adding documents in languages other than English.

To search for documents in a supported language in the console, select your index, then select the option to search your index from the navigation menu. Choose the language that you want to return documents by selecting the search settings and then selecting a language from the dropdown **Language**.

The following examples show how to search for documents in Spanish.

To search an index in Spanish in the console

1. Sign in to the AWS Management Console and open the Amazon Kendra console at http://console.aws.amazon.com/kendra/.
2. In the navigation menu, choose **Indexes** and choose your index.
3. In the navigation menu, choose the option to search your index.
4. In the search settings, select the **Languages** dropdown and choose Spanish.
5. Enter a query into the text box and then press enter.
6. Amazon Kendra returns the results of the search in Spanish.

To search an index in Spanish using the CLI, Python or Java

- The following example searches an index in Spanish. Change the value `searchString` to your search query and the value `indexID` to the identifier of the index that you want to search. The language code for Spanish is `es`. You can replace this with your own language code.

  CLI

  ```
  {
    "EqualsTo":{
      "Key": "_language_code",
      "Value": {
      "StringValue": "es"
      }
    }
  }
  ```

  Python

  ```
  import boto3
  import pprint

  kendra = boto3.client("kendra")

  # Provide the index ID
  index_id = "index-id"
  # Provide the query text
  query = "search-string"
  ```

```
# Includes the index ID, query text, and language attribute filter
response = kendra.query(
        QueryText = query,
        IndexId = index_id,
        AttributeFilter = {
            "EqualsTo": {
                "Key": "_language_code",
                "Value": {
                    "StringValue": "es"
                    }
                }
            })

print ("\nSearch results|Resultados de la búsqueda: " + query + "\n")

for query_result in response["ResultItems"]:

    print("-------------------")
    print("Type: " + str(query_result["Type"]))

    if query_result["Type"]=="ANSWER" or
 query_result["Type"]=="QUESTION_ANSWER":
        answer_text = query_result["DocumentExcerpt"]["Text"]
        print(answer_text)

    if query_result["Type"]=="DOCUMENT":
        if "DocumentTitle" in query_result:
            document_title = query_result["DocumentTitle"]["Text"]
            print("Title: " + document_title)
        document_text = query_result["DocumentExcerpt"]["Text"]
        print(document_text)

    print("------------------\n\n")
```

Java

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.QueryRequest;
import software.amazon.awssdk.services.kendra.model.QueryResponse;
```

```
import software.amazon.awssdk.services.kendra.model.QueryResultItem;

public class SearchIndexExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        String query = "searchString";
        String indexId = "indexID";

        QueryRequest queryRequest = QueryRequest.builder()
            .queryText(query)
            .indexId(indexId)
            .attributeFilter(
                 AttributeFilter.builder()
                     .withEqualsTo(
                         DocumentAttribute.builder()
                             .withKey("_language_code")
                             .withValue("es")
                             .build())
                     .build())
            .build();

        QueryResponse queryResponse = kendra.query(queryRequest);

        System.out.println(String.format("\nSearch results|
                                          Resultados de la búsqueda: %s",
 query));
        for(QueryResultItem item: queryResponse.resultItems()) {
            System.out.println("----------------------");
            System.out.println(String.format("Type: %s", item.type()));

            switch(item.type()) {
                case QUESTION_ANSWER:
                case ANSWER:
                    String answerText = item.documentExcerpt().text();
                    System.out.println(answerText);
                    break;
                case DOCUMENT:
                    String documentTitle = item.documentTitle().text();
                    System.out.println(String.format("Title: %s",
 documentTitle));
                    String documentExcerpt = item.documentExcerpt().text();
                    System.out.println(String.format("Excerpt: %s",
 documentExcerpt));
```

```
                    break;
                default:
                    System.out.println(String.format("Unknown query result type:
 %s", item.type()));

            }

            System.out.println("-----------------------\n");
        }
    }
}
```

Retrieving passages

You can use the Retrieve API as a retriever for retrieval augmented generation (RAG) systems.

RAG systems use generative artificial intelligence to build question-answering applications. RAG systems consist of a retriever and large language models (LLM). Given a query, the retriever identifies the most relevant chunks of text from a corpus of documents and feeds it to the LLM to provide the most useful answer. Then, the LLM analyzes the relevant text chunks or passages and generates a comprehensive response for the query.

The `Retrieve` API looks at chunks of text or excerpts that are referred to as *passages* and returns the top passages that are most relevant to the query.

Like the Query API, the `Retrieve` API also searches for relevant information. Retrieve API's information retrieval takes into account the query's context, and all the available information from the indexed documents. However, by default, the `Query` API only returns excerpt passages of up to 100 token words. With the `Retrieve` API, you can retrieve longer passages of up to 200 token words and up to 100 semantically relevant passages. This doesn't include question-answer or FAQ type responses from your index. The passages, also called chunks, are text excerpts that can be semantically extracted from multiple documents and multiple parts of the same document. Kendra's GenAI Enterprise Edition index offers high accuracy results for retrieve, using a hybrid search over vector and keyword indices along with ranking by deep learning models.

You can also do the following with the `Retrieve` API:

- Override boosting at the index level
- Filter based on document fields or attributes

- Filter based on the user or their group access to documents
- View the confidence score bucket for a retrieved passage result. The confidence bucket provides a relative ranking that indicates how confident Amazon Kendra is that the response is relevant to the query.

> **Note**
>
> Confidence score buckets are currently available only for English.

You can also include certain fields in the response that might provide useful additional information.

The `Retrieve` API currently doesn't support the following features: querying using advance query syntax, suggested spell corrections for queries, faceting, query suggestions to autocomplete search queries, and incremental learning. Any retrieve API queries will not surface in the analytics dashboard.

The `Retrieve` API shares the number of query capacity units that you set for your index. For more information on what's included in a single capacity unit and the default base capacity for an index, see Adjusting capacity.

> **Note**
>
> You can't add capacity if you are using the Amazon Kendra Developer Edition; you can only add capacity when using Amazon Kendra Enterprise Edition. For more information on what's included in the Developer and Enterprise Editions, see Amazon Kendra Editions.

The following is an example of using the `Retrieve` API to retrieve the top 100 most relevant passages from documents in an index for the query "how does amazon kendra work?"

Python

```
import boto3
import pprint

kendra = boto3.client("kendra")
```

```
# Provide the index ID
index_id = "index-id"
# Provide the query text
query = "how does amazon kendra work?"
# You can retrieve up to 100 relevant passages
# You can paginate 100 passages across 10 pages, for example
page_size = 10
page_number = 10

result = kendra.retrieve(
        IndexId = index_id,
        QueryText = query,
        PageSize = page_size,
        PageNumber = page_number)

print("\nRetrieved passage results for query: " + query + "\n")

for retrieve_result in result["ResultItems"]:

    print("-------------------")
    print("Title: " + str(retrieve_result["DocumentTitle"]))
    print("URI: " + str(retrieve_result["DocumentURI"]))
    print("Passage content: " + str(retrieve_result["Content"]))
    print("------------------\n\n")
```

Java

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.RetrieveRequest;
import software.amazon.awssdk.services.kendra.model.RetrieveResult;
import software.amazon.awssdk.services.kendra.model.RetrieveResultItem;

public class RetrievePassageExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        String indxId = "index-id";
        String query = "how does amazon kendra work?";
        Integer pgSize = 10;
        Integer pgNumber = 10;
```

```
        RetrieveRequest retrieveRequest = retrieveRequest
            .builder()
            .indexId(indxId)
            .queryText(query)
            .pageSize(pgSize)
            .pageNumber(pgNumber)
            .build();

        RetrieveResult retrieveResult = kendra.retrieve(retrieveRequest);

        System.out.println(String.format("\nRetrieved passage results for query:
 %s", query));
        for(RetrieveResultItem item: retrieveResult.resultItems()) {
            System.out.println("----------------------");
            System.out.println(String.format("Title: %s", documentTitle));
            System.out.println(String.format("URI: %s", documentURI));
            System.out.println(String.format("Passage content: %s", content));
            System.out.println("-----------------------\n");
        }
    }
}
```

Browsing an index

Note

Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

You can browse documents by their attributes or facets without having to type a search query. Amazon Kendra *Index Browse* can help your users discover documents by freely browsing an index without a specific query in mind. This also helps your users broadly browse an index as a starting point in their search.

Index Browse can only be used for searching by document attribute or facet with a sorting type. You cannot search an entire index using Index Browse. If the query text is missing, then Amazon Kendra asks for a document attribute filter or a facet, and a sorting type.

To allow index browsing using the Query API, you must include AttributeFilter or Facet, and SortingConfiguration. To allow index browsing in the console, select your index under **Indexes** in the navigation menu, then select the option to search your index. In the search box, press the *Enter* key twice. Select the dropdown **Filter search results** to choose a filter and select the dropdown **Sort** to choose a sorting type.

The following is an example of browsing an index for documents in the language Spanish in descending order of document creation date.

CLI

```
aws kendra query \
--index-id "index-id" \
--attribute-filter '{
    "EqualsTo":{
      "Key": "_language_code",
      "Value": {
        "StringValue": "es"
      }
    }
  }' \
--sorting-configuration '{
    "DocumentAttributeKey": "_created_at",
    "SortOrder": "DESC"
  }'
```

Python

```
import boto3

kendra = boto3.client("kendra")

# Must include the index ID, the attribute filter, and sorting configuration
response = kendra.query(
        IndexId = "index-id",
        AttributeFilter = {
            "EqualsTo": {
                "Key": "_language_code",
                "Value": {
                    "StringValue": "es"
                    }
                }
```

```
            },
        SortingConfiguration = {
            "DocumentAttributeKey": "_created_at",
            "SortOrder": "DESC"})

print("\nSearch results|Resultados de la búsqueda: \n")

for query_result in response["ResultItems"]:

    print("-------------------")
    print("Type: " + str(query_result["Type"]))

    if query_result["Type"]=="ANSWER" or query_result["Type"]=="QUESTION_ANSWER":
        answer_text = query_result["DocumentExcerpt"]["Text"]
        print(answer_text)

    if query_result["Type"]=="DOCUMENT":
        if "DocumentTitle" in query_result:
            document_title = query_result["DocumentTitle"]["Text"]
            print("Title: " + document_title)
        document_text = query_result["DocumentExcerpt"]["Text"]
        print(document_text)

    print("------------------\n\n")
```

Java

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.QueryRequest;
import software.amazon.awssdk.services.kendra.model.QueryResult;
import software.amazon.awssdk.services.kendra.model.QueryResultItem;

public class SearchIndexExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();
        QueryRequest queryRequest = QueryRequest.builder()
            .withIndexId("index-id")
            .withAttributeFilter(AttributeFilter.builder()
                .withEqualsTo(DocumentAttribute.builder()
                    .withKey("_language_code")
                    .withValue(DocumentAttributeValue.builder()
```

```
                        .withStringValue("es")
                        .build())
                    .build())
                .build())
            .withSortingConfiguration(SortingConfiguration.builder()
                .withDocumentAttributeKey("_created_at")
                .withSortOrder("DESC")
                .build())
            .build());

        QueryResult queryResult = kendra.query(queryRequest);
        for (QueryResultItem item : queryResult.getResultItems()) {
            System.out.println("----------------------");
            System.out.println(String.format("Type: %s", item.getType()));

            switch (item.getType()) {
                case QueryResultType.QUESTION_ANSWER:
                case QueryResultType.ANSWER:
                    String answerText = item.getDocumentExcerpt().getText();
                    System.out.println(answerText);
                    break;
                case QueryResultType.DOCUMENT:
                    String documentTitle = item.getDocumentTitle().getText();
                    System.out.println(String.format("Title: %s", documentTitle));
                    String documentExcerpt = item.getDocumentExcerpt().getText();
                    System.out.println(String.format("Excerpt: %s",
 documentExcerpt));
                    break;
                default:
                    System.out.println(String.format("Unknown query result type:
 %s", item.getType()));
            }
            System.out.println("----------------------\n");
        }
    }
}
```

Featuring search results

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

You can feature certain documents in the search results when your users issue certain queries. This helps make the results more visible and prominent for your users. Featured results are separated out from the usual list of results, and displayed at the top of the search page. You can experiment with featuring different documents for different queries, or ensure certain documents get the visibility they deserve.

You map specific queries to specific documents for featuring in the results. If a query contains an exact match, then one or more specific documents are featured in the search results.

For example, you can specify that if your users issue the query 'new products 2023', then select the documents titled 'What's new' and 'Coming soon' to feature at the top of the search results page. This helps ensure these documents on new products get the visibility they deserve.

Amazon Kendra doesn't duplicate search results if a result is already selected for featuring at the top of the search results page. A featured result isn't again ranked as the first result if it is already featured above all other results.

In order to feature certain results, you must specify an exact match of a full text query, not a partial match of a query using a keyword or phrase contained within a query. For example, if you only specify the query 'Kendra' in a featured result set, queries such as 'How does Kendra semantically rank results?' will not render the featured results. Featured results are designed for specific queries, rather than queries that are too broad in scope. Amazon Kendra naturally handles keyword type queries to rank the most useful documents in the search results, avoiding excessive featuring of results based on simple keywords.

If there are certain queries that your users frequently use, then you can specify these queries for featured results. For example, if you look at your top queries using Amazon Kendra Analytics and find that specific queries, such as 'How does kendra semantically rank results?' and 'kendra semantic search', are frequently used, then these queries might be useful to specify for featuring the document titled 'Amazon Kendra search 101'.

Amazon Kendra treats queries for featured results as case insensitive. Amazon Kendra converts a query to lower case, and replaces trailing white space characters with a single space. Amazon Kendra matches all other characters as they are when you specify your queries for featured results.

You create a set of featured results that you map to certain queries using the CreateFeaturedResultsSet API. If you use the console, you select your index and then select **Featured results** in the navigation menu to create a featured results set. You can create up to 50 sets of featured results per index, up to four documents to be featured per set, and up to 49 query texts per featured results set. You can request to increase these limits by contacting Support.

You can select the same document across multiple sets of featured results. However, you must not use the same exact match query text across multiple sets. The queries you specify for featured results must be unique per featured results set for each index.

You can arrange the order of documents when selecting up to four featured documents. If you use the API, the order you list the featured documents is the same as displayed in the featured results. If you use the console, you can simply drag and drop the order of documents when you select documents for featuring in the results.

Access control, where certain users and groups have access to certain documents and others don't, is still honored when configuring featured results. That's also true for user context filtering. For example, user A belongs to the 'Interns' company group, which shouldn't access documents on company secrets. If user A enters a query that features a company secret document, user A doesn't see this document featured in their results. That's also true for any other results on the search results page. You can also use tags to control access to a featured results set, which is an Amazon Kendra resource for which you control access.

The following is an example of creating a set of featured results with the queries "new products 2023", "new products available" mapped to the documents titled "What's new" (doc-id-1) and "Coming soon" (doc-id-2).

CLI

```
aws kendra create-featured-results-set \
 --featured-results-set-name 'New product docs to feature' \
 --description "Featuring What's new and Coming soon docs" \
 --index-id index-id \
 --query-texts 'new products 2023' 'new products available' \
 --featured-documents '{"Id":"doc-id-1", "Id":"doc-id-2"}'
```

Python

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Create a featured results set.")

# Provide a name for the featured results set
featured_results_name = "New product docs to feature"
# Provide an optional decription for the featured results set
description = "Featuring What's new and Coming soon docs"
# Provide the index ID for the featured results set
index = "index-id"
# Provide a list of query texts for the featured results set
queries = ['new products 2023', 'new products available']
# Provide a list of document IDs for the featured results set
featured_doc_ids = [{"Id":"doc-id-1"}, {"Id":"doc-id-2"}]

try:
    featured_results_set_response = kendra.create_featured_results_set(
        FeaturedResultsSetName = featured_results_name,
        Decription = description,
        Index = index,
        QueryTexts = queries,
        FeaturedDocuments = featured_doc_ids
    )

    pprint.pprint(featured_results_set_response)

    featured_results_set_id = featured_results_set_response["FeaturedResultsSetId"]

    while True:
        # Get the details of the featured results set, such as the status
        featured_results_set_description = kendra.describe_featured_results_set(
            Id = featured_results_set_id
        )
        status = featured_results_set_description["Status"]
        print(" Featured results set status: "+status)

except  ClientError as e:
```

```
        print("%s" % e)

print("Program ends.")
```

Tabular search for HTML

Note

Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

Amazon Kendra's tabular search feature can search and extract answers from tables embedded in HTML documents. When you search your index, Amazon Kendra includes an excerpt from a table if it's relevant to the query and provides useful information.

Amazon Kendra looks at all of the information within the body text of a document, including useful information in tables. For example, an index contains business reports with tables on operation costs, income, and other financial information. For the query, "what is the annual operation cost from 2020-2022?", Amazon Kendra can return an excerpt from a table that contains relevant table columns "Operations (millions USD)" and "Financial year", and table rows containing income values for 2020, 2021, and 2022. The table excerpt is included in the result, along with the document title, a link to the full document, and any other document fields you choose to include.

Table excerpts can be displayed in the search results whether the information is found in one cell of a table or multiple cells. For example, Amazon Kendra can display a table excerpt tailored to each of these kinds of queries:

- "highest interest rate credit card in 2020"
- "highest interest rate credit card from 2020-2022"
- "top 3 highest interest rate credit cards in 2020-2022"
- "credit cards with interest rates less than 10%"
- "all available low interest credit cards"

Amazon Kendra highlights the table cell or cells that are most relevant to the query. The most relevant cells with their corresponding rows, columns and column names are displayed in the search result. The table excerpt displays up to five columns and three rows, depending on how many table cells are relevant to the query and how many columns are available in the original table. The top most relevant cell is displayed in the table excerpt, along with the next most relevant cells.

The response includes the confidence bucket (`MEDIUM`, `HIGH`, `VERY_HIGH`) to show how relevant the table answer is to the query. If a table cell value is `VERY_HIGH` in confidence, then it becomes the 'top answer' and is highlighted. For table cell values that are `HIGH` in confidence, then they are highlighted. For table cell values that are `MEDIUM` in confidence, then they are not highlighted. The overall confidence for the table answer is returned in the response. For example, if a table contains mostly table cells with `HIGH` confidence, then the overall confidence returned in the response for the table answer is `HIGH` confidence.

By default, tables aren't given a higher level of importance or more weight than other components of a document. Within a document, if a table is only slightly relevant to a query, but there's a highly relevant paragraph, Amazon Kendra returns an excerpt of the paragraph. Search results display the piece of content that provides the best possible answer and most useful information, in the same document or other documents. If the confidence for a table falls below `MEDIUM` confidence, then the table excerpt is not returned in the response.

To use tabular search on an existing index, you must re-index your content.

Amazon Kendra tabular search supports synonyms (including custom synonyms). Amazon Kendra only supports documents in English with HTML tables that are within the table tag.

The following example shows table excerpt included in the query result. To view a sample JSON with query responses, including table excerpts, see Query responses and types.

Python

```
import boto3
import pprint

kendra = boto3.client("kendra")

# Provide the index ID
index_id = <index-id>
# Provide the query text
query = "search string"

response = kendra.query(
```

```
        QueryText = query,
        IndexId = index_id)

print("\nSearch results for query: " + query + "\n")

for query_result in response["ResultItems"]:

    print("-------------------")
    print("Type: " + str(query_result["Type"]))
    print("Type: " + str(query_result["Format"]))

    if query_result["Type"]=="ANSWER" and query_result["Format"]=="TABLE":
        answer_table = query_result["TableExcerpt"]
        print(answer_table)

    if query_result["Type"]=="ANSWER" and query_result["Format"]=="TEXT":
        answer_text = query_result["DocumentExcerpt"]
        print(answer_text)

    if query_result["Type"]=="QUESTION_ANSWER":
        question_answer_text = query_result["DocumentExcerpt"]["Text"]
        print(question_answer_text)

    if query_result["Type"]=="DOCUMENT":
        if "DocumentTitle" in query_result:
            document_title = query_result["DocumentTitle"]["Text"]
            print("Title: " + document_title)
        document_text = query_result["DocumentExcerpt"]["Text"]
        print(document_text)

    print("------------------\n\n")
```

Java

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.QueryRequest;
import software.amazon.awssdk.services.kendra.model.QueryResponse;
import software.amazon.awssdk.services.kendra.model.QueryResultItem;

public class SearchIndexExample {
    public static void main(String[] args) {
```

```
        KendraClient kendra = KendraClient.builder().build();

        String query = "search string";
        String indexId = "index-id";

        QueryRequest queryRequest = QueryRequest
            .builder()
            .queryText(query)
            .indexId(indexId)
            .build();

        QueryResponse queryResponse = kendra.query(queryRequest);

        System.out.println(String.format("\nSearch results for query: %s", query));
        for(QueryResultItem item: queryResponse.resultItems()) {
            System.out.println("----------------------");
            System.out.println(String.format("Type: %s", item.type()));
            System.out.println(String.format("Format: %s", item.format()));

            switch(item.format()) {
                case TABLE:
                    String answerTable = item.TableExcerpt();
                    System.out.println(answerTable);
                    break;
            }

            switch(item.format()) {
                case TEXT:
                    String answerText = item.DocumentExcerpt();
                    System.out.println(answerText);
                    break;
            }

            switch(item.type()) {
                case QUESTION_ANSWER:
                    String questionAnswerText = item.documentExcerpt().text();
                    System.out.println(questionAnswerText);
                    break;
                case DOCUMENT:
                    String documentTitle = item.documentTitle().text();
                    System.out.println(String.format("Title: %s", documentTitle));
                    String documentExcerpt = item.documentExcerpt().text();
                    System.out.println(String.format("Excerpt: %s",
  documentExcerpt));
```

```
                        break;
                    default:
                        System.out.println(String.format("Unknown query result type:
 %s", item.type()));

                }

                System.out.println("-----------------------\n");
            }
        }
    }
```

Query suggestions

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

Amazon Kendra *Query suggestions* can help your users type their search queries faster and guide their search.

Amazon Kendra suggests queries relevant to your users based on one of the following:

- Popular queries in the query history or query log
- The contents of document fields/attributes

You can set your preference for using the query history or document fields by setting `SuggestionTypes` as either `QUERY` or `DOCUMENT_ATTRIBUTES` and calling GetQuerySuggestions. By default, Amazon Kendra uses the query history to base suggestions on. If the query history and document fields are both activated when you call UpdateQuerySuggestionsConfig and you haven't set your `SuggestionTypes` preference to use document fields, then Amazon Kendra uses the query history.

If you use the console, you can base query suggestions on either the query history or document fields. You first select your index and then select **Query suggestions** under **Enrichments** in the navigation menu. Then select **Configure query suggestions**. After you configure query suggestions, you are directed to a search console where you can select either the **Query history** or **Document fields** in the right panel and enter a search query in the search bar.

By default, query suggestions using the query history and document fields are both activated at no additional cost. You can deactivate these types of query suggestions at any time by using the `UpdateQuerySuggestionsConfig` API. To deactivate query suggestions based on the query history, set `Mode` to `DISABLED` when calling `UpdateQuerySuggestionsConfig`. To deactivate query suggestions based on document fields, set `AttributeSuggestionsMode` to `INACTIVE` in the document fields configuration and then call `UpdateQuerySuggestionsConfig>`. If you use the console, you can deactivate query suggestions in the **Query suggestions settings**.

Query suggestions are case insensitive. Amazon Kendra converts the query prefix and the suggested query to lower case, ignores all single and double quotation marks, and replaces multiple white space characters with a single space. Amazon Kendra matches all other special characters as they are. Amazon Kendra does not show any suggestions if a user types fewer than two characters or more than 60 characters.

Topics

- Query suggestions using query history
- Query suggestions using document fields
- Block certain queries or document field content from suggestions

Query suggestions using query history

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

Topics

- Settings for selecting queries for suggestions
- Clear suggestions while retaining query history
- No suggestions available

You can choose to suggest queries relevant to your users based on popular queries in the query history or query log. Amazon Kendra uses all of the queries that your users search for and learns from these queries to make suggestions to your users. Amazon Kendra suggests popular queries to users when they start typing their query. Amazon Kendra suggests a query if the prefix or first few characters of the query matches what the user starts typing as their query.

For example, a user starts typing the query 'upcoming events'. Amazon Kendra has learned from the query history that many users have searched for 'upcoming events 2050' many times. The user sees 'upcoming events 2050' appear directly underneath their search bar, autocompleting their search query. The user selects this query suggestion, and the document 'New events: What's happening in 2050' is returned in the search results.

You can specify how Amazon Kendra selects eligible queries to suggest to your users. For example, you can specify that a query suggestion must have been searched by at least 10 unique users (default is three), have been searched within the last 30 days, and does not contain any words or phrases from your block list. Amazon Kendra requires that a query has at least one search result and contains at least one word of more than four characters.

Settings for selecting queries for suggestions

You can configure the following settings for selecting queries for suggestions by using the UpdateQuerySuggestionsConfig API:

- **Mode**—Query suggestions using the query history are either `ENABLED` or `LEARN_ONLY`. Amazon Kendra activates query suggestions by default. `LEARN_ONLY` turns off query suggestions. If turned off, Amazon Kendra continues to learn suggestions but doesn't make query suggestions to users.
- **Query log time window**—How recent your queries are in your query log time window. The time window is an integer value for the number of days from current day to past days.
- **Queries without user information**—Set to `TRUE` to include all queries, or set to `FALSE` to only include queries with user information. You can use this setting if your search application includes user information, such as the user ID, when a user issues a query. By default, this setting doesn't filter out queries if there's no specific user information associated with the queries. However, you can use this setting to only make suggestions based on queries that include user information.
- **Unique users**—The minimum number of unique users who must search a query for the query to be eligible to suggest to your users. This number is an integer value.
- **Query count**—The minimum number of times a query must be searched for the query to be eligible to suggest to your users. This number is an integer value.

These settings affect how queries are selected as popular queries to suggest to your users. How you tune your settings will depend on your specific needs, for example:

- If your users usually search once a month on average, then you can set the number of days in the query log time window to 30 days. By using that setting, you capture most of your users' recent queries before they become outdated in the time window.
- If only a small number of your queries include user information, and you don't want to suggest queries based on a small sample size, then you can set queries to include all users.
- If you define popular queries as being searched by at least 10 unique users and searched at least 100 times, then you set the unique users to 10 and the query count to 100.

> **Warning**
>
> Your changes to settings might not take effect immediately. You can track the settings changes by using the DescribeQuerySuggestionsConfig API. The time for your updated settings to take effect depends on the updates that you make and the number of search queries in your index. Amazon Kendra automatically updates suggestions every 24 hours, after you change a setting or after you apply a block list.

CLI

To retrieve query suggestions

```
aws kendra get-query-suggestions \
 --index-id index-id \
 --query-text "query-text" \
 --suggestion-types '["QUERY"]' \
 --max-suggestions-count 1 // If you want to limit the number of suggestions
```

To update query suggestions

For example, to change the query log time window and the minimum number of times a query must be searched:

```
aws kendra update-query-suggestions-config \
 --index-id index-id \
 --query-log-look-back-window-in-days 30 \
```

```
 --minimum-query-count 100
```

Python

To retrieve query suggestions

```
import boto3
from botocore.exceptions import ClientError

kendra = boto3.client("kendra")

print("Get query suggestions.")

# Provide the index ID
index_id = "index-id"

# Provide the query text
query_text = "query"

# Provide the query suggestions type
query_suggestions_type = "QUERY"


# If you want to limit the number of suggestions
num_suggestions = 1

try:
    query_suggestions_response = kendra.get_query_suggestions(
        IndexId = index_id,
        QueryText = query_text,
        SuggestionTypes = query_suggestions_type,
        MaxSuggestionsCount = num_suggestions
    )

    # Print out the suggestions you received
    if ("Suggestions" in query_suggestions_response.keys()) {
        for (suggestion: query_suggestions_response["Suggestions"]) {
            print(suggestion["Value"]["Text"]["Text"]);
        }
    }

except ClientError as e:
        print("%s" % e)
```

```
print("Program ends.")
```

To update query suggestions

For example, to change the query log time window and the minimum number of times a query must be searched:

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Updating query suggestions settings/configuration for an index.")

# Provide the index ID
index_id = "index-id"

# Configure the settings you want to update
minimum_query_count = 100
query_log_look_back_window_in_days = 30

try:
    kendra.update_query_suggestions_config(
        IndexId = index_id,
        MinimumQueryCount = minimum_query_count,
        QueryLogLookBackWindowInDays = query_log_look_back_window_in_days
    )

    print("Wait for Amazon Kendra to update the query suggestions.")

    while True:
        # Get query suggestions description of settings/configuration
        query_sugg_config_response = kendra.describe_query_suggestions_config(
            IndexId = index_id
        )

        # If status is not UPDATING, then quit
        status = query_sugg_config_response["Status"]
        print(" Updating query suggestions config. Status: " + status)
        if status != "UPDATING":
            break
```

```
        time.sleep(60)

except ClientError as e:
        print("%s" % e)

print("Program ends.")
```

Clear suggestions while retaining query history

Note

Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

You can clear query suggestions by using the ClearQuerySuggestions API. Clearing suggestions deletes existing query suggestions only, not the queries in the query history. When you clear suggestions, Amazon Kendra learns new suggestions based on new queries added to the query log from the time you cleared suggestions.

CLI

To clear query suggestions

```
aws  kendra clear-query-suggestions \
 --index-id index-id
```

Python

To clear query suggestions

```
import boto3
from botocore.exceptions import ClientError

kendra = boto3.client("kendra")

print("Clearing out query suggestions for an index.")

# Provide the index ID
index_id = "index-id"
```

```
try:
    kendra.clear_query_suggestions(
        IndexId = index_id
    )

    # Confirm last cleared date-time and that there are no suggestions
    query_sugg_config_response = kendra.describe_query_suggestions_config(
        IndexId = index_id
    )
    print("Query Suggestions last cleared at: " +
 str(query_sugg_config_response["LastClearTime"]));
    print("Number of suggestions available from the time of clearing: " +
 str(query_sugg_config_response["TotalSuggestionsCount"]));

except ClientError as e:
        print("%s" % e)

print("Program ends.")
```

No suggestions available

If you don't see suggestions for a query, it could be for one of the following reasons:

- There are not enough queries in your index for Amazon Kendra to learn from.
- Your query suggestions settings are too strict, resulting in most queries being filtered out from suggestions.
- You recently cleared suggestions, and Amazon Kendra still needs time for new queries to accumulate in order to learn new suggestions.

You can check your current settings using the DescribeQuerySuggestionsConfig API.

Query suggestions using document fields

Topics

- Settings for selecting fields for suggestions
- User control in document fields

You can choose to suggest queries relevant to your users based on the contents of document fields. Instead of using the query history to suggest other popular relevant queries, you can use information contained within a document field that is useful to autocompleting the query. Amazon Kendra looks for relevant content in fields set to `Suggestable` and that closely aligns with your user's query. Then, Amazon Kendra suggests this content to your user when they start typing their query.

For example, if you specify the title field to base suggestions on and a user starts typing the query 'How amazon ken...', the most relevant title 'How Amazon Kendra works' could be suggested to autocomplete the search. The user sees 'How Amazon Kendra works' appear directly underneath their search bar, autocompleting their search query. The user selects this query suggestion, and the document 'How Amazon Kendra works' is returned in the search results.

You can use the contents of any document field of `String` and `StringList` type to suggest a query by setting the field to `Suggestable` as part of your fields configuration for query suggestions. You can also use a block list so that suggested document fields that contain certain words or phrases are not shown to your users. You can use one block list. The block list applies whether you set query suggestions to use the query history or document fields.

Settings for selecting fields for suggestions

You can configure the following settings for selecting document fields for suggestions using AttributeSuggestionsConfig and calling the UpdateQuerySuggestionsConfig API to update the settings at the index level:

- **Field/attribute suggestions mode**—Query suggestions using document fields are either `ACTIVE` or `INACTIVE`. Amazon Kendra activates query suggestions by default.
- **Suggestible fields/attributes**—The field names or field keys to base suggestions on. These fields must be set to `TRUE` for `Suggestable`, as part of the fields configuration. You can override the fields configuration at the query level while maintaining the configuration at the index level. Use the GetQuerySuggestions API to change `AttributeSuggestionConfig` at the query level. This configuration at the query level can be useful for quickly experimenting with using different document fields without having to update the configuration at the index level.
- **Additional fields/attributes**—The additional fields that you want to include in the response for a query suggestion. These fields are used to provide extra information in the response; however, they are not used to base suggestions on.

Warning

Your changes to settings might not take effect immediately. You can track the settings changes by using the DescribeQuerySuggestionsConfig API. The time for your updated settings to take effect depends on the updates that you make. Amazon Kendra automatically updates suggestions every 24 hours, after you change a setting or after you apply a block list.

CLI

To retrieve query suggestions and override the document fields configuration at the query level instead of having to change the configuration at the index level.

```
aws kendra get-query-suggestions \
 --index-id index-id \
 --query-text "query-text" \
 --suggestion-types '["DOCUMENT_ATTRIBUTES"]' \
 --attribute-suggestions-config '{"SuggestionAttributes":'["field/attribute key
 1", "field/attribute key 2"]', "AdditionalResponseAttributes":'["response field/
attribute key 1", "response field/attribute key 2"]'}' \
 --max-suggestions-count 1 // If you want to limit the number of suggestions
```

To update query suggestions

For example, to change the document fields configuration at the index level:

```
aws kendra update-query-suggestions-config \
 --index-id index-id \
 --attribute-suggestions-config '{"SuggestableConfigList": '[{"SuggestableConfig":
 "_document_title", "Suggestable": true}]', "AttributeSuggestionsMode": "ACTIVE"}'
```

Python

To retrieve query suggestions and override the document fields configuration at the query level instead of having to change the configuration at the index level.

```
import boto3
from botocore.exceptions import ClientError

kendra = boto3.client("kendra")
```

```
print("Get query suggestions.")

# Provide the index ID
index_id = "index-id"

# Provide the query text
query_text = "query"

# Provide the query suggestions type
query_suggestions_type = "DOCUMENT_ATTRIBUTES"

# Override fields/attributes configuration at query level
configuration = {"SuggestionAttributes":
    '["field/attribute key 1", "field/attribute key 2"]',
      "AdditionalResponseAttributes":
          '["response field/attribute key 1", "response field/attribute key 2"]'
          }

# If you want to limit the number of suggestions
num_suggestions = 1

try:
    query_suggestions_response = kendra.get_query_suggestions(
        IndexId = index_id,
        QueryText = query_text,
        SuggestionTypes = [query_suggestions_type],
        AttributeSuggestionsConfig = configuration,
        MaxSuggestionsCount = num_suggestions
    )

    # Print out the suggestions you received
    if ("Suggestions" in query_suggestions_response.keys()) {
        for (suggestion: query_suggestions_response["Suggestions"]) {
            print(suggestion["Value"]["Text"]["Text"]);
        }
    }

except ClientError as e:
        print("%s" % e)

print("Program ends.")
```

To update query suggestions

For example, to change the document fields configuration at the index level:

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Updating query suggestions settings/configuration for an index.")

# Provide the index ID
index_id = "index-id"

# Configure the settings you want to update at the index level
configuration = {"SuggestableConfigList":
    '[{"SuggestableConfig": "_document_title", "Suggestable": true}]',
       "AttributeSuggestionsMode": "ACTIVE"
       }

try:
    kendra.update_query_suggestions_config(
        IndexId = index_id,
        AttributeSuggestionsConfig = configuration
    )

    print("Wait for Amazon Kendra to update the query suggestions.")

    while True:
        # Get query suggestions description of settings/configuration
        query_sugg_config_response = kendra.describe_query_suggestions_config(
            IndexId = index_id
        )

        # If status is not UPDATING, then quit
        status = query_sugg_config_response["Status"]
        print(" Updating query suggestions config. Status: " + status)
        if status != "UPDATING":
            break
        time.sleep(60)

except ClientError as e:
```

```
        print("%s" % e)

print("Program ends.")
```

User control in document fields

You can apply user context filtering to the document fields that you want to base query suggestions on. This filters document field information based on the user or their group access to documents. For example, an intern searches the company's portal and doesn't have access to a top-secret company document. Therefore, suggested queries based on the top-secret document's title, or any other suggestible field, is not shown to the intern.

You can index your documents with an access control list (ACL), defining which users and groups are assigned access to which documents. Then, you can apply user context filtering to your documents fields for query suggestions. User context filtering that is currently set for your index is the same user context filtering applied to your document fields configuration for query suggestions. User context filtering is part of your document fields configuration. You use the AttributeSuggestionsGetConfig and call GetQuerySuggestions.

Block certain queries or document field content from suggestions

A *block list* stops Amazon Kendra from suggesting certain queries to your users. A block list is a list of words or phrases that you want to exclude from query suggestions. Amazon Kendra excludes queries containing an exact match of the words or phrases in the block list.

You can use a block list to safeguard against offensive words or phrases that commonly appear in your query history or document fields and that Amazon Kendra could select as suggestions. A block list can also prevent Amazon Kendra from suggesting queries that contain information that is not ready to be publicly released or announced. For example, your users frequently query about an upcoming release of a potential new product. However, you don't want to suggest the product because you're not ready to release it.You can block queries that contain the product name and product information from suggestions.

You can create a block list for queries by using the CreateQuerySuggestionsBlockList API. You put each block word or phrase on a separate line in a text file. Then you upload the text file to your Amazon S3 bucket and provide the path or location to the file in Amazon S3. Amazon Kendra currently supports creating only one block list.

You can replace the text file of your blocked words and phrases in your Amazon S3 bucket. To update the block list in Amazon Kendra, use the UpdateQuerySuggestionsBlockList API.

Use the DescribeQuerySuggestionsBlockList API to get the status of your block list. `DescribeQuerySuggestionsBlockList` can also provide you with other useful information, such as the following:

- When your block list was last updated
- How many words or phrases are in your current block list
- Helpful error messages when creating a block list

You can also use the ListQuerySuggestionsBlockLists API to get a list of block list summaries for an index.

To delete your block list, use the DeleteQuerySuggestionsBlockList API.

Your updates to the block list might not take effect right away. You can track updates by using the `DescribeQuerySuggestionsBlockList` API.

CLI

To create a block list

```
aws kendra create-query-suggestions-block-list \
 --index-id index-id \
 --name "block-list-name" \
 --description "block-list-description" \
 --source-s3-path "Bucket=bucket-name,Key=query-suggestions/block_list.txt" \
 --role-arn role-arn
```

To update a block list

```
aws kendra update-query-suggestions-block-list \
 --index-id index-id \
 --name "new-block-list-name" \
 --description "new-block-list-description" \
 --source-s3-path "Bucket=bucket-name,Key=query-suggestions/new_block_list.txt" \
 --role-arn role-arn
```

To delete a block list

```
aws kendra delete-query-suggestions-block-list \
 --index-id index-id \
 --id block-list-id
```

Python

To create a block list

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Create a query suggestions block list.")

# Provide a name for the block list
block_list_name = "block-list-name"
# Provide an optional description for the block list
block_list_description = "block-list-description"
# Provide the IAM role ARN required for query suggestions block lists
block_list_role_arn = "role-arn"

# Provide the index ID
index_id = "index-id"

s3_bucket_name = "bucket-name"
s3_key = "query-suggestions/block_list.txt"
source_s3_path = {
    'Bucket': s3_bucket_name,
    'Key': s3_key
}

try:
    block_list_response = kendra.create_query_suggestions_block_list(
        Description = block_list_description,
        Name = block_list_name,
        RoleArn = block_list_role_arn,
        IndexId = index_id,
        SourceS3Path = source_s3_path
    )
```

```
    print(block_list_response)

    block_list_id = block_list_response["Id"]

    print("Wait for Amazon Kendra to create the block list.")

    while True:
        # Get block list description
        block_list_description = kendra.describe_query_suggestions_block_list(
            Id = block_list_id,
            IndexId = index_id
        )
        # If status is not CREATING, then quit
        status = block_list_description["Status"]
        print("Creating block list. Status: " + status)
        if status != "CREATING":
            break
        time.sleep(60)

except ClientError as e:
        print("%s" % e)

print("Program ends.")
```

To update a block list

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Update a block list for query suggestions.")

# Provide the block list name you want to update
block_list_name = "new-block-list-name"
# Provide the block list description you want to update
block_list_description = "new-block-list-description"
# Provide the IAM role ARN required for query suggestions block lists
block_list_role_arn = "role-arn"

# Provide the block list ID
```

```
block_list_id = "block-list-id"
# Provide the index ID
index_id = "index-id"

s3_bucket_name = "bucket-name"
s3_key = "query-suggestions/new_block_list.txt"
source_s3_path = {
'Bucket': s3_bucket_name,
'Key': s3_key
}

try:
    kendra.update_query_suggestions_block_list(
        Id = block_list_id,
        IndexId = index_id,
        Description = block_list_description,
        Name = block_list_name,
        RoleArn = block_list_role_arn,
        SourceS3Path = source_s3_path
    )

    print("Wait for Amazon Kendra to update the block list.")

    while True:
        # Get block list description
        block_list_description = kendra.describe_query_suggestions_block_list(
            Id = block_list_id,
            IndexId = index_id
        )
        # If status is not UPDATING, then the update has finished
        status = block_list_description["Status"]
        print("Updating block list. Status: " + status)
        if status != "UPDATING":
            break
        time.sleep(60)

except ClientError as e:
print("%s" % e)

print("Program ends.")
```

To delete a block list

```
import boto3
from botocore.exceptions import ClientError

kendra = boto3.client("kendra")

print("Delete a block list for query suggestions.")

# provide the block list ID
query_suggestions_block_list_id = "query-suggestions-block-list-id"
# Provide the index ID
index_id = "index-id"

try:
    kendra.delete_query_suggestions_block_list(
        Id = query_suggestions_block_list_id,
        IndexId = index_id
    )

except ClientError as e:
        print("%s" % e)

print("Program ends.")
```

Query spell checker

Note

Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

Amazon Kendra *Spell Checker* suggests spell corrections for a query. This can help you keep occurrences of zero search results to a minimum and return relevant results. Your users might receive zero search results from misspelled queries with no matching results or no returned documents. Or, your users might receive irrelevant search results from misspelled queries.

Spell Checker is designed to suggest corrections for misspelled words based on words that appear in your indexed documents and how closely a corrected word matches a misspelled word. For example, if the word 'statements' appears in your indexed documents, then this could closely match the misspelled word 'statments' in the query 'year-end financial statments'.

Spell Checker returns the intended or corrected words that replace misspelled words in the original query text. For example, 'depoying kendre search' could return 'deploying Kendra search' You can also use offset locations provided in the API to highlight or italicize the returned corrected words in a query in your front-end application. In the console, the corrected words are highlighted or italicized by default. For example, '*deploying Kendra* search'.

For business-specific or specialized terms that appear in your indexed documents, Spell Checker does not misunderstand these terms as spellings mistakes in the query. For example, 'amazon macie' is not corrected to 'amazon mace'.

For hyphenated words, such as 'year-end', Spell Checker treats these as individual words to suggest corrections for these words. For example, the suggested correction for 'yaer-end' could be 'year-end'.

For `DOCUMENT` and `QUESTION_ANSWER` query response types, Spell Checker suggests corrections to misspelled words based on words in the document body. The document body is more reliable than the title for suggesting corrections that closely match the misspelled words. For `ANSWER` query response types, Spell Checker suggests corrections based on words in the default question and answer document in your index.

You can activate Spell Checker using the SpellCorrectionConfiguration object. You set `IncludeQuerySpellCheckSuggestions` to `TRUE`. Spell Checker is activated by default in the console. It is built into the console by default.

Spell Checker can also suggest spell corrections for queries in multiple languages, not only English. For a list of languages supported for Spell Checker, see Amazon Kendra supported languages.

Using the query spell checker with default limits

Spell Checker is designed with certain defaults or limits. The following is a list of current limits that apply when you activate spell correction suggestions.

- Suggested spell corrections cannot be returned for words that are less than three characters or more than 30 characters in length. To allow for more than 30 characters or less than three characters, contact Support.
- Suggested spell corrections cannot restrict suggestions based on user access control or your access control list for user context filtering. Spell corrections are based on all words in your

indexed documents, whether the words are restricted to certain users or not. If you want to avoid certain words appearing in the suggested spell corrections for queries, then do not activate `SpellCorrectionConfiguration`.

- Suggested spell corrections cannot be returned for words that include numbers. For example, 'how 2 not br8k ubun2'.
- Suggested spell corrections cannot use words that don't appear in your indexed documents.
- Suggested spell corrections cannot use words that are frequented less than 0.01 percent in your indexed documents. To change the 0.01% threshold, contact Support.

Filtering and facet search

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

You can improve the search results or response from the Query API by using filters. Filters restrict the documents in the response to ones that directly apply to the query. To create faceted search suggestions, use Boolean logic to filter out specific document attributes from the response or documents that don't match specific criteria. You can specify facets using the `Facets` parameter in the `Query` API.

To search documents that you have indexed with Amazon Kendra for Amazon Lex, use AMAZON.KendraSearchIntent. For an example of configuring Amazon Kendra with Amazon Lex, see Creating a FAQ Bot for an Amazon Kendra Index. You can also provide a filter for the response by using AttributeFilter. This is the query filter in JSON when configuring `AMAZON.KendraSearchIntent`. To provide an attribute filter when configuring a search intent in the console, go to the intent editor and choose Amazon Kendra query to provide a query filter in JSON. For more information about `AMAZON.KendraSearchIntent`, see the Amazon Lex documentation guide.

Facets

Facets are scoped views of a set of search results. For example, you can provide search results for cities across the world, where documents are filtered by a specific city with which they are associated. Or, you can create facets to display results by a specific author.

You can use a document attribute or metadata field associated with a document as a facet so that your users can search by categories or values within that facet. You can also display nested facets in the search results so that your users can search not only by a category or field but also by a sub category or sub field.

The following example shows how to get facet information for the "City" custom attribute.

```
response=kendra.query(
        QueryText = query,
        IndexId = index,
        Facets = [
            {
                "DocumentAttributeKey" : "City"
            }
        ]
        )
```

You can use nested facets to further narrow the search. For example, the document attribute or facet "City" includes a value called "Seattle". In addition, the document attribute or facet "CityRegion" includes the values "North" and "South" for documents assigned to "Seattle". You can display nested facets with their counts in the search results so that documents can be searched not only by city but also by a region within a city.

Note that nested facets could impact query latency. A general rule is the more nested facets that you use, the greater potential impact on latency. Other factors that affect latency include the average size of documents indexed, the size of your index, highly complex queries, and the overall load on your Amazon Kendra index.

The following example shows how to get facet information for the "CityRegion" custom attribute, as a nested facet within "City".

```
response=kendra.query(
        QueryText = query,
        IndexId = index,
        Facets = [
            {
                "DocumentAttributeKey" : "City",
                "Facets": [
                    {
                        "DocumentAttributeKey" : "CityRegion"
                    }
```

```
                ]
            }
        ]
        )
```

Facet information, such as the document count, is returned in the `FacetResults` response array. You use the contents to display faceted search suggestions in your application. For example, if the document attribute "City" contains the city that a search could apply to, use that information to display a list of city searches. Users can choose a city to filter their search results. To make the faceted search, call the [Query](#) API and use the chosen document attribute to filter the results.

You can display up to 10 facet values per facet for a query, and only one nested facet within a facet. If you want to increase these limits, contact [Support](#). If you want to limit the number of facet values per facet to less than 10, you can specify this in the `Facet` object.

The following sample JSON response shows facets scoped to the "City" document attribute. The response includes the count of documents for the facet value.

```
{
    'FacetResults': [
        {
            'DocumentAttributeKey': 'City',
            'DocumentAttributeValueCountPairs': [
                {
                    'Count': 3,
                    'DocumentAttributeValue': {
                        'StringValue': 'Dubai'
                    }
                },
                {
                    'Count': 3,
                    'DocumentAttributeValue': {
                        'StringValue': 'Seattle'
                    }
                },
                {
                    'Count': 1,
                    'DocumentAttributeValue': {
                        'StringValue': 'Paris'
                    }
                }
            ]
```

```
        }
    ]
```

You can also display facet information for a nested facet, such as a region within a city, to further filter the search results.

The following sample JSON response shows facets scoped to the "CityRegion" document attribute, as a nested facet within "City". The response includes the count of documents for the nested facet values.

```
{
    'FacetResults': [
        {
            'DocumentAttributeKey': 'City',
            'DocumentAttributeValueCountPairs': [
                {
                    'Count': 3,
                    'DocumentAttributeValue': {
                        'StringValue': 'Dubai'
                    },
                    'FacetResults': [
                        {
                            'DocumentAttributeKey': 'CityRegion',
                            'DocumentAttributeValueCountPairs': [
                                {
                                    'Count': 2,
                                    'DocumentAttributeValue': {
                                        'StringValue': 'Bur Dubai'
                                    }
                                },
                                {
                                    'Count': 1,
                                    'DocumentAttributeValue': {
                                        'StringValue': 'Deira'
                                    }
                                }
                             ]
                        }
                    ]
                },
                {
                    'Count': 3,
                    'DocumentAttributeValue': {
```

```
                    'StringValue': 'Seattle'
                },
                'FacetResults': [
                    {
                        'DocumentAttributeKey': 'CityRegion',
                        'DocumentAttributeValueCountPairs': [
                            {
                                'Count': 1,
                                'DocumentAttributeValue': {
                                    'StringValue': 'North'
                                }
                            },
                            {
                                'Count': 2,
                                'DocumentAttributeValue': {
                                    'StringValue': 'South'
                                }
                            }
                        ]
                    }
                ]
            },
            {
                'Count': 1,
                'DocumentAttributeValue': {
                    'StringValue': 'Paris'
                },
                'FacetResults': [
                    {
                        'DocumentAttributeKey': 'CityRegion',
                        'DocumentAttributeValueCountPairs': [
                            {
                                'Count': 1,
                                'DocumentAttributeValue': {
                                    'StringValue': 'City center'
                                }
                            }
                        ]
                    }
                ]
            }
        }
    ]
```

```
}
```

When you use a string list field to create facets, the facet results returned are based on the contents of the string list. For example, if you have a string list field that contains two items, one with the list "dachshund", "sausage dog" and one with the value "husky", you get `FacetResults` with three facets.

For more information, see [Query responses and response types](#).

Using document attributes to filter search results

By default, `Query` returns all search results. To filter responses, you can perform logical operations on the document attributes. For example, if you only want documents for a specific city, you can filter on the "City" and "State" custom document attributes. You use [AttributeFilter](#) to create a Boolean operation on filters that you supply.

Most attributes can be used to filter responses for all [response types](#). However, the `_excerpt_page_number` attribute is only applicable to `ANSWER` response types when filtering responses.

The following example shows how to perform a logical AND operation by filtering on a specific city, *Seattle*, and state, *Washington*.

```
response=kendra.query(
        QueryText = query,
        IndexId = index,
        AttributeFilter = {'AndAllFilters':
            [
                {"EqualsTo": {"Key": "City","Value": {"StringValue": "Seattle"}}},
                {"EqualsTo": {"Key": "State","Value": {"StringValue": "Washington"}}}
            ]
            }
        )
```

The following example shows how to perform a logical OR operation for when any of the `Fileformat`, `Author`, or `SourceURI` keys match the specified values.

```
response=kendra.query(
        QueryText = query,
```

```
        IndexId = index,
        AttributeFilter = {'OrAllFilters':
            [
                {"EqualsTo": {"Key": "Fileformat","Value": {"StringValue":
 "AUTO_DETECT"}}},
                {"EqualsTo": {"Key": "Author","Value": {"StringValue": "Ana
 Carolina"}}},
                {"EqualsTo": {"Key": "SourceURI","Value": {"StringValue": "https://
aws.amazonaws.com/234234242342"}}}
            ]
            }
        )
```

For `StringList` fields, use the `ContainsAny` or `ContainsAll` attribute filters to return documents with the specified string. The following example shows how to return all documents that have the values "Seattle" or "Portland" in their `Locations` custom attribute.

```
response=kendra.query(
        QueryText = query,
        IndexId = index,
        AttributeFilter = {
                "ContainsAny": { "Key": "Locations", "Value": { "StringListValue":
 [ "Seattle", "Portland"] }}
            }
        )
```

Filtering each document's attributes in the search results

Amazon Kendra returns document attributes for each document in the search results. You can filter certain document attributes you want to include in the response as part of the search results. By default, all document attributes assigned to a document are returned in the response.

In the following example, only the `_source_uri` and `_author` document attributes are included in the response for a document.

```
response=kendra.query(
        QueryText = query,
        IndexId = index,
        RequestedDocumentAttributes = ["_source_uri", "_author"]
        )
```

Filtering on user context

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

You can filter a user's search results based on the user or their group access to documents. You can use a user token, user ID, or user attribute to filter documents.

User context filtering is a kind of personalized search with the benefit of controlling access to documents. For example, not all teams that search the company portal for information should access top-secret company documents, nor are these documents relevant to all users. Only specific users or groups of teams given access to top-secret documents should see these documents in their search results.

When a document is indexed into Amazon Kendra, a corresponding access control list (ACL) is ingested for most documents. The ACL specifies which user names and group names are allowed or denied access to the document. Documents without an ACL are public documents.

Amazon Kendra can extract the user or group information associated with each document for most data sources. For example, a document in Quip can include a 'share' list of select users that are given access to the document. If you use an S3 bucket as a data source, you provide a JSON file for your ACL and include the S3 path to this file as part of the data source configuration. If you add documents directly to an index, you specify the ACL in the Principal object as part of the document object in the BatchPutDocument API.

If you're using an Amazon Kendra Enterprise or Developer Edition index, you can use the CreateAccessControlConfiguration API to re-configure your existing document level access control without indexing all of your documents again. For example, your index contains top-secret company documents that only certain employees or users should access. One of these users leaves the company or switches to a team that should be blocked from accessing top-secret documents. The user still has access to top-secret documents because the user had access when your documents were previously indexed. You can create a specific access control configuration for the user with deny access. You can later update the access control configuration to allow access in the case the user returns to the company and re-joins the 'top-secret' team. You can re-configure access control for your documents as circumstances change.

To apply your access control configuration to certain documents, you call the BatchPutDocument API with the `AccessControlConfigurationId` included in the Document object.
If you use an S3 bucket as a data source, you update the `.metadata.json` with the `AccessControlConfigurationId` and synchronize your data source. Amazon Kendra currently only supports access control configuration for S3 data sources and documents indexed using the `BatchPutDocument` API.

Filtering by user token

> **Important**
>
> Amazon Kendra GenAI Enterprise Edition indices don't support token-based user access control.

When you query an index, you can use a user token to filter search results based on the user or their group access to documents. When you issue a query, Amazon Kendra extracts and validates the token, pulls and checks the user and group information, and runs the query. All of the documents the user has access to, including public documents, are returned. For more information, see Token-based user access control.

You provide the user token in the UserContext object and pass this in the Query API.

The following shows how to include a user token.

```
response = kendra.query(
    QueryText = query,
    IndexId = index,
    UserToken = {
        Token = "token"
    })
```

You can map users to groups. When you use user-context filtering, it is not required to include all of the groups that a user belongs to when you issue the query. With the PutPrincipalMapping API, you can map users to their groups. If you do not want to use the `PutPrincipalMapping` API, you must provide the user name and all the groups the user belongs to when you issue a query. You can also fetch access levels of groups and users in your IAM Identity Center identity source by using the UserGroupResolutionConfiguration object.

Filtering by user ID and group

When you query an index, you can use the user ID and group to filter search results based on the user or their group access to documents. When you issue a query, Amazon Kendra checks the user and group information and runs the query. All of the documents relevant to the query that the user has access to, including public documents, are returned.

You can also filter search results by data sources that users and groups have access to. Specifying a data source is useful if a group is tied to multiple data sources, but you only want the group to access documents of a certain data source. For example, the groups "Research", "Engineering", and "Sales and Marketing" are all tied to the company's documents stored in the data sources Confluence and Salesforce. However, "Sales and Marketing" team only needs access to customer-related documents stored in Salesforce. So when sales and marketing users search for customer-related documents, they can see documents from Salesforce in their results. Users who do not work in sales and marketing do not see Salesforce documents in their search results.

You provide the user, groups and data sources information in the UserContext object and pass this in the Query API. The user ID, and the list of groups and data sources should match the name you specify in the Principal object to identify the user, groups, and data sources. With the `Principal` object, you can add a user, group, or data source to either an allow list or a deny list for accessing a document.

You are required to provide one of the following:

- User and groups information, and (optional) data sources information.
- Only the user information if you map your users to groups and data sources using the PutPrincipalMapping API. You can also fetch access levels of groups and users in your IAM Identity Center identity source by using the UserGroupResolutionConfiguration object.

If this information is not included in the query, Amazon Kendra returns all documents. If you provide this information, only documents with matching user IDs, groups, and data sources are returned.

The following shows how to include user ID, groups, and data sources.

```
response = kendra.query(
    QueryText = query,
    IndexId = index,
```

```
    UserContext={'Token': 'string', 'UserId': 'string', 'Groups': [ 'string', ],
 'DataSourceGroups': [ { 'GroupId': 'string', 'DataSourceId': 'string' }, ] },)
```

Filtering by user attribute

When you query an index, you can use built-in attributes `_user_id` and `_group_id` to filter search results based on the user and their group access to documents. You can set up to 100 group identifiers. When you issue a query, Amazon Kendra checks the user and group information and runs the query. All documents relevant to the query that the user has access to, including public documents, are returned.

You provide the user and group attributes in the AttributeFilter object and pass this in the Query API.

The following example shows a request that filters the query response based on the user ID and the groups "HR" and "IT", which the user belongs to. The query will return any document that has the user or the "HR" or "IT" groups in the allow list. If the user or either group is in the deny list for a document, the document is not returned.

```
response = kendra.query(
        QueryText = query,
        IndexId = index,
        AttributeFilter = {
            "OrAllFilters": [
                {
                    "EqualsTo": {
                        "Key": "_user_id",
                        "Value": {
                            "StringValue": "user1"
                        }
                     }
                },
                {
                    "EqualsTo": {
                        "Key": "_group_ids",
                        "Value": {
                            "StringListValue": ["HR", "IT"]
                        }
                    }
                }
            ]
```

```
        }
        )
```

You can also specify which data source a group can access in the `Principal` object.

> **Note**
>
> User context filtering isn't an authentication or authorization control for your content. It doesn't do user authentication on the user and groups sent to the `Query` API. It is up to your application to ensure that the user and group information sent to `Query` API is authenticated and authorized.

There is an implementation of user context filtering for each data source. The following section describes each implementation.

Topics

- User context filtering for documents added directly to an index
- User context filtering for frequently asked questions
- User context filtering for data sources

User context filtering for documents added directly to an index

When you add documents directly to an index using the BatchPutDocument API, Amazon Kendra gets user and group information from the `AccessControlList` field of the document. You provide an access control list (ACL) for your documents and the ACL is ingested with your documents.

You specify the ACL in the Principal object as part of the Document object in the `BatchPutDocument` API. You provide the following information:

- The access that the user or group should have. You can say `ALLOW` or `DENY`.
- The type of entity. You can say `USER` or `GROUP`.
- The name of the user or group.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for frequently asked questions

When you add a FAQ to an index, Amazon Kendra gets user and group information from the `AccessControlList` object/field of the FAQ JSON file. You can also use a FAQ CSV file with custom fields or attributes for access control.

You provide the following information:

- The access that the user or group should have. You can say `ALLOW` or `DENY`.
- The type of entity. You can say `USER` or `GROUP`.
- The name of the user or group.

For more information, see FAQ files.

User context filtering for data sources

Amazon Kendra also crawls user and group access control list (ACL) information from supported data source connectors. This is useful for user context filtering, where search results are filtered based on the user or their group access to documents.

> **Important**
>
> Amazon Kendra GenAI Enterprise Edition indices support only v2.0 Amazon Kendra data source connectors.

Topics

User context filtering for Adobe Experience Manager data sources

When you use an Adobe Experience Manager data source, Amazon Kendra gets the user and group information from the Adobe Experience Manager instance.

The group and user IDs are mapped as follows:

- `_group_ids`—Group IDs exist in Adobe Experience Manager content where there are set access permissions. They are mapped from the names of the groups in Adobe Experience Manager.
- `_user_id`—User IDs exist in Adobe Experience Manager content where there are set access permissions. They are mapped from the user emails as the IDs in Adobe Experience Manager.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Alfresco data sources

When you use an Alfresco data source, Amazon Kendra gets the user and group information from the Alfresco instance.

The group and user IDs are mapped as follows:

- `_group_ids`—Group IDs exist in Alfresco on files where there are set access permissions. They are mapped from the system names of the groups (not display names) in Alfresco.
- `_user_id`—User IDs exist in Alfresco on files where there are set access permissions. They are mapped from the user emails as the IDs in Alfresco.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Aurora (MySQL) data sources

When you use a Aurora (MySQL) data source, Amazon Kendra gets user and group information from a column in the source table. You specify this column in the console or using the [TemplateConfiguration](#) object as part of the [CreateDataSource](#) API.

A Aurora (MySQL) database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be a string containing a semicolon delimited list of groups.

User context filtering for Aurora (PostgreSQL) data sources

When you use a Aurora (PostgreSQL) data source, Amazon Kendra gets user and group information from a column in the source table. You specify this column in the console or using the TemplateConfiguration object as part of the CreateDataSource API.

A Aurora (PostgreSQL) database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be a string containing a semicolon delimited list of groups.

User context filtering for Amazon FSx data sources

When you use an Amazon FSx data source, Amazon Kendra gets user and group information from the directory service of the Amazon FSx instance.

The Amazon FSx group and user IDs are mapped as follows:

- `_group_ids`—Group IDs exist in Amazon FSx on files where there are set access permissions. They are mapped from the system group names in the directory service of Amazon FSx.
- `_user_id`—User IDs exist in Amazon FSx on files where there are set access permissions. They are mapped from the system user names in the directory service of Amazon FSx.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for database data sources

When you use a database data source, such as Amazon Aurora PostgreSQL, Amazon Kendra gets user and group information from a column in the source table. You specify this column in the AclConfiguration object as part of the DatabaseConfiguration object in the CreateDataSource API.

A database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be a string containing a semicolon delimited list of groups.

User context filtering for Amazon RDS (Microsoft SQL Server) data sources

When you use a Amazon RDS (Microsoft SQL Server) data source, Amazon Kendra gets user and group information from a column in the source table. You specify this column in the console or using the TemplateConfiguration object as part of the CreateDataSource API.

A Amazon RDS (Microsoft SQL Server) database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be a string containing a semicolon delimited list of groups.

User context filtering for Amazon RDS (MySQL) data sources

When you use a Amazon RDS (MySQL) data source, Amazon Kendra gets user and group information from a column in the source table. You specify this column in the console or using the TemplateConfiguration object as part of the CreateDataSource API.

A Amazon RDS (MySQL) database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be a string containing a semicolon delimited list of groups.

User context filtering for Amazon RDS (Oracle) data sources

When you use a Amazon RDS (Oracle) data source, Amazon Kendra gets user and group information from a column in the source table. You specify this column in the console or using the TemplateConfiguration object as part of the CreateDataSource API.

A Amazon RDS (Oracle) database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be a string containing a semicolon delimited list of groups.

User context filtering for Amazon RDS (PostgreSQL) data sources

When you use a Amazon RDS (PostgreSQL) data source, Amazon Kendra gets user and group information from a column in the source table. You specify this column in the console or using the TemplateConfiguration object as part of the CreateDataSource API.

A Amazon RDS (PostgreSQL) database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be a string containing a semicolon delimited list of groups.

User context filtering for Amazon S3 data sources

You add user context filtering to a document in an Amazon S3 data source using a metadata file associated with the document. You add the information to the `AccessControlList` field in the JSON document. For more information about adding metadata to the documents indexed from an Amazon S3 data source, see S3 document metadata.

You provide three pieces of information:

- The access that the entity should have. You can say `ALLOW` or `DENY`.
- The type of entity. You can say `USER` or `GROUP`.
- The name of the entity.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Box data sources

When you use a Box data source, Amazon Kendra gets user and group information from the Box instance.

The Box group and user IDs are mapped as follows:

- `_group_ids`—Group IDs exist in Box on files where there are set access permissions. They are mapped from the names of the groups in Box.
- `_user_id`—User IDs exist in Box on files where there are set access permissions. They are mapped from the user emails as the user IDs in Box.

You can add up to 200 entries in the AccessControlList field.

User context filtering for Confluence data sources

When you use a Confluence data source, Amazon Kendra gets user and group information from the Confluence instance.

You configure user and group access to spaces using the space permissions page. For pages and blogs, you use the restrictions page. For more information about space permissions, see Space Permissions Overview on the Confluence Support website. For more information about page and blog restrictions, see Page Restrictions on the Confluence Support website.

The Confluence group and user names are mapped as follows:

- _group_ids—Group names are present on spaces, pages, and blogs where there are restrictions. They are mapped from the name of the group in Confluence. Group names are always lower case.

- _user_id—User names are present on the space, page, or blog where there are restrictions. They are mapped depending on the type of Confluence instance that you are using.

  For Confluence connector v1.0

  - Server—The _user_id is the user name. The username is always lower case.
  - Cloud—The _user_id is the account ID of the user.

  For Confluence connector v2.0

  - Server—The _user_id is the user name. The username is always lower case.
  - Cloud—The _user_id is the email ID of the user.

> **Important**
>
> For user context filtering to work correctly for your Confluence connector, you need to make sure that the visibility of a user granted access to a Confluence page is set to **Anyone**. For more information, see Set your email visibility in Atlassian Developer Documentation.

You can add up to 200 entries in the AccessControlList field.

User context filtering for Dropbox data sources

When you use a Dropbox data source, Amazon Kendra gets the user and group information from the Dropbox instance.

The group and user IDs are mapped as follows:

- `_group_ids`—Group IDs exist in Dropbox on files where there are set access permissions. They are mapped from the names of the groups in Dropbox.
- `_user_id`—User IDs exist in Dropbox on files where there are set access permissions. They are mapped from the user emails as the IDs in Dropbox.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Drupal data sources

When you use a Drupal data source, Amazon Kendra gets the user and group information from the Drupalinstance.

The group and user IDs are mapped as follows:

- `_group_ids` – Group IDs exist in Drupal on files where there are set access permissions. They are mapped from the names of the groups in Drupal.
- `_user_id` – User IDs exist in Drupal on files where there are set access permissions. They are mapped from the user emails as the IDs in Drupal.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for GitHub data sources

When you use a GitHub data source, Amazon Kendra gets user information from the GitHub instance.

The GitHub user IDs are mapped as follows:

- `_user_id`—User IDs exist in GitHub on files where there are set access permissions. They are mapped from the user emails as the IDs in GitHub.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Gmail data sources

When you use a Gmail data source, Amazon Kendra gets the user information from the Gmail instance.

The user IDs are mapped as follows:

- `_user_id` – User IDs exist in Gmail on files where there are set access permissions. They are mapped from the user emails as the IDs in Gmail.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Google Drive data sources

A Google Workspace Drive data source returns user and group information for Google Drive users and groups. Group and domain membership are mapped to the `_group_ids` index field. The Google Drive user name is mapped to the `_user_id` field.

When you provide one or more user email addresses in the `Query` API, only documents that have been shared with those email addresses are returned. The following `AttributeFilter` parameter only returns documents shared with "martha@example.com".

```
"AttributeFilter": {
                "EqualsTo":{
                   "Key": "_user_id",
                   "Value": {
                       "StringValue": "martha@example.com"
                   }
               }
           }
```

If you provide one or more group email addresses in the query, only documents shared with the groups are returned. The following `AttributeFilter` parameter only returns documents shared with the "hr@example.com" group.

```
"AttributeFilter": {
                "EqualsTo":{
                   "Key": "_group_ids",
                   "Value": {
                       "StringListValue": ["hr@example.com"]
                   }
```

```
            }
        }
```

If you provide the domain in the query, all documents shared with the domain are returned. The following `AttributeFilter` parameter returns documents shared with the "example.com" domain.

```
"AttributeFilter": {
                "EqualsTo":{
                   "Key": "_group_ids",
                   "Value": {
                       "StringListValue": ["example.com"]
                   }
               }
           }
```

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for IBM DB2 data sources

When you use a IBM DB2 data source, Amazon Kendra gets user and group information from a column in the source table. You specify this column in the console or using the [TemplateConfiguration](#) object as part of the [CreateDataSource](#) API.

A IBM DB2 database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be a string containing a semicolon delimited list of groups.

User context filtering for Jira data sources

When you use a Jira data source, Amazon Kendra gets user and group information from the Jira instance.

The Jira user IDs are mapped as follows:

- `_user_id`—User IDs exist in Jira on files where there are set access permissions. They are mapped from the user emails as the user IDs in Jira.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Microsoft Exchange data sources

When you use a Microsoft Exchange data source, Amazon Kendra gets the user information from the Microsoft Exchange instance.

The Microsoft Exchange user IDs are mapped as follows:

- `_user_id`—User IDs exist in Microsoft Exchange permissions for users to access certain content. They are mapped from the user names as the IDs in Microsoft Exchange.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Microsoft OneDrive data sources

Amazon Kendra retrieves user and group information from Microsoft OneDrive when it indexes the documents on the site. The user and group information is taken from the underlying Microsoft SharePoint site that hosts OneDrive.

When you use a OneDrive user or group for filtering search results, calculate the ID as follows:

1. Get the site name. For example, `https://host.onmicrosoft.com/sites/siteName.`
2. Take the MD5 hash of the site name. For example, `430a6b90503eef95c89295c8999c7981`.
3. Create the user email or group ID by concatenating the MD5 hash with a vertical bar (|) and the ID. For example, if a group name is "localGroupName", the group ID would be:

   ```
   "430a6b90503eef95c89295c8999c7981 | localGroupName"
   ```

   > **Note**
   >
   > Include a space before and after the vertical bar. The vertical bar is used to identify `localGroupName` with its MD5 hash.

   For the user name "someone@host.onmicrosoft.com," the user ID would be the following:

   ```
   "430a6b90503eef95c89295c8999c7981 | someone@host.onmicrosoft.com"
   ```

Send the user or group ID to Amazon Kendra as the `_user_id` or `_group_id` attribute when you call the Query API. For example, the AWS CLI command that uses a group to filter the search results looks like this:

```
aws kendra  query \
                --index-id index ID
                --query-text "query text"
                --attribute-filter '{
                   "EqualsTo":{
                     "Key": "_group_id",
                     "Value": {"StringValue": "430a6b90503eef95c89295c8999c7981 |
 localGroupName"}
                  }}'
```

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Microsoft OneDrive v2.0 data sources

A Microsoft OneDrive v2.0 data source returns section and page information from OneDrive access control list (ACL) entities. Amazon Kendra uses the OneDrive tenant domain to connect to the OneDrive instance and then can filter search results based on user or group access to sections and file names.

For standard objects, the `_user_id` and `_group_id` are used as follows:

- `_user_id`— Your Microsoft OneDrive user email ID is mapped to the `_user_id` field.
- `_group_id`— Your Microsoft OneDrive group email is mapped to the `_group_id` field.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Microsoft SharePoint data sources

Amazon Kendra retrieves user and group information from Microsoft SharePoint when it indexes the site documents. To filter search results based on user or group access, provide user and group information when you call the `Query` API.

To filter using a user name, use the user's email address. For example, johnstiles@example.com.

When you use a SharePoint group for filtering search results, calculate the group ID as follows:

For local groups

1. Get the site name. For example, `https://host.onmicrosoft.com/sites/siteName.`
2. Take the SHA256 hash of the site name. For example, `430a6b90503eef95c89295c8999c7981`.
3. Create the group ID by concatenating the SHA256 hash with a vertical bar (|) and the group name. For example, if the group name is "localGroupName", the group ID would be:

   ```
   "430a6b90503eef95c89295c8999c7981 | localGroupName"
   ```

> **Note**
>
> Include a space before and after the vertical bar. The vertical bar is used to identify `localGroupName` with its SHA256 hash.

Send the group ID to Amazon Kendra as the `_group_id` attribute when you call the [Query API](). For example, the AWS CLI command looks like this:

```
aws kendra  query \
                --index-id index ID
                --query-text "query text"
                --attribute-filter '{
                   "EqualsTo":{
                     "Key": "_group_id",
                     "Value": {"StringValue": "430a6b90503eef95c89295c8999c7981 |
 localGroupName"}
                  }}'
```

For AD groups

1. Use the AD group ID for configuring filtering of search results.

Send the group ID to Amazon Kendra as the `_group_id` attribute when you call the [Query]() API. For example, the AWS CLI command looks like this:

```
aws kendra  query \
                --index-id index ID
                --query-text "query text"
                --attribute-filter '{
                   "EqualsTo":{
```

```
                    "Key": "_group_id",
                    "Value": {"StringValue": "AD group"}
                }}'
```

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Microsoft SQL Server data sources

When you use a Microsoft SQL Server data source, Amazon Kendra gets user and group information from a column in the source table. You specify this column in the console or using the TemplateConfiguration object as part of the CreateDataSource API.

A Microsoft SQL Server database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be a string containing a semicolon delimited list of groups.

User context filtering for Microsoft Teams data sources

Amazon Kendra retrieves user information from Microsoft Teams when it indexes the documents. The user information is taken from the underlying Microsoft Teams instance.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Microsoft Yammer data sources

Amazon Kendra retrieves user information from Microsoft Yammer when it indexes the documents. The user and group information is taken from the underlying Microsoft Yammer instance.

The Microsoft Yammer user IDs are mapped as follows:

- `_email_id`— The Microsoft email ID mapped to the `_user_id` field.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for MySQL data sources

When you use a MySQL data source, Amazon Kendra gets user and group information from a column in the source table. You specify this column in the console or using the TemplateConfiguration object as part of the CreateDataSource API.

A MySQL database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be a string containing a semicolon delimited list of groups.

User context filtering for Oracle Database data sources

When you use a Oracle Database data source, Amazon Kendra gets user and group information from a column in the source table. You specify this column in the console or using the TemplateConfiguration object as part of the CreateDataSource API.

A Oracle Database database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be a string containing a semicolon delimited list of groups.

User context filtering for PostgreSQL data sources

When you use a PostgreSQL data source, Amazon Kendra gets user and group information from a column in the source table. You specify this column in the console or using the TemplateConfiguration object as part of the CreateDataSource API.

A PostgreSQL database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be a string containing a semicolon delimited list of groups.

User context filtering for Quip data sources

When you use a Quip data source, Amazon Kendra gets the user information from the Quip instance.

The Quip user IDs are mapped as follows:

- `_user_id`—User IDs exist in Quip on files where there are set access permissions. They are mapped from the user emails as the IDs in Quip.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Salesforce data sources

A Salesforce data source returns user and group information from Salesforce access control list (ACL) entities. You can apply user context filtering to Salesforce standard objects and chatter feeds. User context filtering is not available for Salesforce knowledge articles.

If you map any Salesforce field to Amazon Kendra document title and document body fields, Amazon Kendra will use data from the document title and body fields in search responses.

For standard objects, the `_user_id` and `_group_ids` are used as follows:

- `_user_id`—The user name of the Salesforce user.
- `_group_ids`—
  - Name of the Salesforce `Profile`
  - Name of the Salesforce `Group`
  - Name of the Salesforce `UserRole`
  - Name of the Salesforce `PermissionSet`

For chatter feeds, the `_user_id` and `_group_ids` are used as follows:

- `_user_id`—The user name of the Salesforce user. Only available if the item is posted in the user's feed.
- `_group_ids`—Group IDs are used as follows. Only available if the feed item is posted in a chatter or collaboration group.
  - The name of the chatter or collaboration group.

- If the group is public, `PUBLIC:ALL`.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for ServiceNow data sources

User context filtering for ServiceNow is supported only for the TemplateConfiguration API and ServiceNow Connector v2.0. ServiceNowConfiguration API and ServiceNow Connector v1.0. do not support user context filtering.

When you use a ServiceNow data source, Amazon Kendra gets the user and group information from the ServiceNow instance.

The group and user IDs are mapped as follows:

- `_group_ids`—Group IDs exist in ServiceNow on files where there are set access permissions. They are mapped from the role names of `sys_ids` in ServiceNow.
- `_user_id`—User IDs exist in ServiceNow on files where there are set access permissions. They are mapped from the user emails as the IDs in ServiceNow.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Slack data sources

When you use a Slack data source, Amazon Kendra gets the user information from the Slack instance.

The Slack user IDs are mapped as follows:

- `_user_id`—User IDs exist in Slack on messages and channels where there are set access permissions. They are mapped from the user emails as the IDs in Slack.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Zendesk data sources

When you use a Zendesk data source, Amazon Kendra gets the user and group information from the Zendesk instance.

The group and user IDs are mapped as follows:

- `_group_ids`—Group IDs exist in Zendesk tickets and articles where there are set access permissions. They are mapped from the names of the groups in Zendesk.
- `_user_id`—Group IDs exist in Zendesk tickets and articles where there are set access permissions. They are mapped from the user emails as the IDs in Zendesk.

You can add up to 200 entries in the `AccessControlList` field.

Query responses and response types

Note

Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

Amazon Kendra supports different query responses and response types.

Query responses

A call to the Query API returns information about the results of a search. The results are in an array ofQueryResultItem objects (`ResultItems`). Each `QueryResultItem` includes a summary of the result. Document attributes associated with the query result are included.

Summary information

The summary information varies depending on the type of result. In each case, it includes document text that matches the search term. It also includes highlight information that you can use to highlight the search text in your application's output. For example, if the search term is *what is the height of the Space Needle?*, the summary information includes text location for the words *height* and *space needle*. For information about response types, see Query responses and response types.

Document attributes

Each result contains document attributes for the document that matches a query. Some of the attributes are predefined, such as `DocumentId`, `DocumentTitle`, and `DocumentUri`. Others are custom attributes that you define. You can use document attributes to filter the response from the `Query` API. For example, you might want only the documents written by a specific author or a specific version of a document. For more information, see [Filtering and facet search](). You specify document attributes when you add documents to an index. For more information, see [Custom fields or attributes]().

The following is sample JSON code for a query result. Note the document attributes in `DocumentAttributes` and `AdditionalAttributes`.

```
{
    "QueryId": "query-id",
    "ResultItems": [
        {
            "Id": "result-id",
            "Type": "ANSWER",
            "AdditionalAttributes": [
                {
                    "Key": "AnswerText",
                    "ValueType": "TEXT_WITH_HIGHLIGHTS_VALUE",
                    "Value": {
                        "TextWithHighlightsValue": {
                            "Text": "text",
                            "Highlights": [
                                {
                                    "BeginOffset": 55,
                                    "EndOffset": 90,
                                    "TopAnswer": false
                                }
                            ]
                        }
                    }
                }
            ],
            "DocumentId": "document-id",
            "DocumentTitle": {
                "Text": "title"
            },
            "DocumentExcerpt": {
                "Text": "text",
                "Highlights": [
                    {
                        "BeginOffset": 0,
                        "EndOffset": 300,
                        "TopAnswer": false
```

```
                }
            ]
        },
        "DocumentURI": "uri",
        "DocumentAttributes": [],
        "ScoreAttributes": "score",
        "FeedbackToken": "token"
    },
    {
        "Id": "result-id",
        "Type": "ANSWER",
        "Format": "TABLE",
        "DocumentId": "document-id",
        "DocumentTitle": {
            "Text": "title"
        },
        "TableExcerpt": {
            "Rows": [{
                "Cells": [{
                    "Header": true,
                    "Highlighted": false,
                    "TopAnswer": false,
                    "Value": "value"
                }, {
                    "Header": true,
                    "Highlighted": false,
                    "TopAnswer": false,
                    "Value": "value"
                }, {
                    "Header": true,
                    "Highlighted": false,
                    "TopAnswer": false,
                    "Value": "value"
                }, {
                    "Header": true,
                    "Highlighted": false,
                    "TopAnswer": false,
                    "Value": "value"
                }]
            }, {
                "Cells": [{
                    "Header": false,
                    "Highlighted": false,
                    "TopAnswer": false,
```

```
                        "Value": "value"
                    }, {
                        "Header": false,
                        "Highlighted": false,
                        "TopAnswer": false,
                        "Value": "value"
                    }, {
                        "Header": false,
                        "Highlighted": true,
                        "TopAnswer": true,
                        "Value": "value"
                    }, {
                        "Header": false,
                        "Highlighted": false,
                        "TopAnswer": false,
                        "Value": "value"
                    ]}
                  }],
                    "TotalNumberofRows": number
    },
            "DocumentURI": "uri",
            "ScoreAttributes": "score",
            "FeedbackToken": "token"
        },
        {
            "Id": "result-id",
            "Type": "DOCUMENT",
            "AdditionalAttributes": [],
            "DocumentId": "document-id",
            "DocumentTitle": {
                "Text": "title",
                "Highlights": []
            },
            "DocumentExcerpt": {
                "Text": "text",
                "Highlights": [
                    {
                        "BeginOffset": 74,
                        "EndOffset": 77,
                        "TopAnswer": false
                    }
                ]
            },
            "DocumentURI": "uri",
```

```
                "DocumentAttributes": [
                    {
                        "Key": "_source_uri",
                        "Value": {
                            "StringValue": "uri"
                        }
                    }
                ],
                "ScoreAttributes": "score",
                "FeedbackToken": "token",
            }
        ],
        "FacetResults": [],
        "TotalNumberOfResults": number
}
```

Response types

Amazon Kendra returns three types of query response.

- Answer (includes table answer)
- Document
- Question and answer

The type of the response is returned in the `Type` response field of the [QueryResultItem](#) object.

Answer

Amazon Kendra detected one or more question answers in the response. A factoid is the response to a who, what, when, or where question such as *Where is the nearest service center to me?* Amazon Kendra returns text in the index that best matches the query. The text is in the `AnswerText` field and contains highlight information for the search term within the response text. `AnswerText` includes the full document excerpt with highlighted text, while `DocumentExcerpt` includes the truncated (290 characters) document excerpt with highlighted text.

Amazon Kendra only returns one answer per document, and that is the answer with the highest confidence. To return multiple answers from a document, you must split the document into multiple documents.

```
{
```

```
    'AnswerText': {
        'TextWithHighlights': [
            {
                'BeginOffset': 271,
                'EndOffset': 279,
                'TopAnswer': False
            },
            {
                'BeginOffset': 481,
                'EndOffset': 489,
                'TopAnswer': False
            },
            {
                'BeginOffset': 547,
                'EndOffset': 555,
                'TopAnswer': False
            },
            {
                'BeginOffset': 764,
                'EndOffset': 772,
                'TopAnswer': False
            }
        ],
        'Text': 'Asynchronousoperationscan\n''alsoprocess
\n''documentsthatareinPDF''format.UsingPDFformatfilesallowsyoutoprocess''multi-
page\n''documents.\n''Forinformationabouthow''AmazonTextractrepresents
\n''documentsasBlockobjects,
        ''seeDocumentsandBlockObjects.
\n''\n''\n''\n''Forinformationaboutdocument''limits,
        seeLimitsinAmazonTextract.
\n''\n''\n''\n''TheAmazonTextractsynchronous''operationscanprocessdocumentsstoredinanAmazon
\n''S3Bucketoryoucanpass''base64encodedimagebytes.\n''Formoreinformation,

 see''CallingAmazonTextractSynchronousOperations.''Asynchronousoperationsrequireinputdocuments
\n''tobesuppliedinanAmazon''S3Bucket.'
    },
    'DocumentExcerpt': {
        'Highlights': [
            {
                'BeginOffset': 0,
                'EndOffset': 300,
                'TopAnswer': False
            }
        ],
```

```
        'Text': 'Asynchronousoperationscan\n''alsoprocess
\n''documentsthatareinPDF''format.UsingPDFformatfilesallowsyoutoprocess''multi-page
\n''documents.\n''ForinformationabouthowAmazon''Textractrepresents\n'''
    },
    'Type': 'ANSWER'
}
```

Document

Amazon Kendra returns ranked documents for those that match the search term. The ranking is based on the confidence that Amazon Kendra has in the accuracy of the search result. Information about the matching document is returned in the QueryResultItem. It includes the title of the document. The excerpt includes highlight information for search text and the section of matching text in the document. The URI for matching documents is in the `SourceURI` document attribute. The following sample JSON shows the document summary for a matching document.

```
{
    'DocumentTitle': {
        'Highlights': [
            {
                'BeginOffset': 7,
                'EndOffset': 15,
                'TopAnswer': False
            },
            {
                'BeginOffset': 97,
                'EndOffset': 105,
                'TopAnswer': False
            }
        ],
        'Text': 'AmazonTextractAPIPermissions: Actions,
        \n''Permissions,
        andResourcesReference-''AmazonTextract'
    },
    'DocumentExcerpt': {
        'Highlights': [
            {
                'BeginOffset': 68,
                'EndOffset': 76,
                'TopAnswer': False
            },
            {
```

```
                'BeginOffset': 121,
                'EndOffset': 129,
                'TopAnswer': False
            }
        ],
        'Text': '...LoggingandMonitoring\tMonitoring
\n''\tCloudWatchMetricsforAmazonTextract
\n''\tLoggingAmazonTextractAPICallswithAWSCloudTrail\n''\tAPIReference\tActions
\tAnalyzeDocument\n''\tDetectDocumentText\n''\tGetDocumentAnalysis...'
    },
    'Type': 'DOCUMENT'
}
```

Question and answer

A question and answer response is returned when Amazon Kendra matches a question with one of the frequently asked questions in your index. The response includes the matching question and answer in the QueryResultItem field. It also includes highlight information for query terms detected in query string. The following JSON shows a question and answer response. Note that the response includes the question text.

```
{
    'AnswerText': {
        'TextWithHighlights': [

        ],
        'Text': '605feet'
    },
    'DocumentExcerpt': {
        'Highlights': [
            {
                'BeginOffset': 0,
                'EndOffset': 8,
                'TopAnswer': False
            }
        ],
        'Text': '605feet'
    },
    'Type': 'QUESTION_ANSWER',
    'QuestionText': {
        'Highlights': [
            {
                'BeginOffset': 12,
```

```
                    'EndOffset': 18,
                    'TopAnswer': False
                },
                {
                    'BeginOffset': 26,
                    'EndOffset': 31,
                    'TopAnswer': False
                },
                {
                    'BeginOffset': 32,
                    'EndOffset': 38,
                    'TopAnswer': False
                }
            ],
            'Text': 'whatistheheightoftheSpaceNeedle?'
        }
}
```

For information about adding question and answer text to an index, see Creating FAQ.

Tuning and sorting responses

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

You can modify the effect of a field or attribute on the search relevance through relevance tuning. You can also sort the search results by a certain attribute or field.

Topics

- Tuning responses
- Sorting responses

Tuning responses

You can modify the effect of a field or attribute on the search relevance through relevance tuning. To quickly test relevance tuning, use the Query API to pass in tuning configurations in the query.

Then you can see the different search results that you get from different configurations. Relevance tuning at the query level is not supported in the console. You can also tune fields or attributes that are of the type `StringList` at the index level only. For more information, see Tuning search relevance.

By default, query responses are sorted by the relevance score that Amazon Kendra determines for each result in the response.

You can tune results for any built-in or custom attribute/field of the following types:

- Date value
- Long value
- String value

You can't sort attributes of the following type:

- String list values

Rank and tune document results (AWS SDK)

Set the `Searchable` parameter to true to boost the document metadata configuration.

To tune an attribute in a query, set the `DocumentRelevanceOverrideConfigurations` parameter of the `Query` API and specify the name of the attribute to tune.

The following JSON example shows a `DocumentRelevanceOverrideConfigurations` object that overrides the tuning for the attribute called "department" in the index.

```
"DocumentRelevanceOverrideConfigurations" : [
    "Name": "department",
    "Relevance": {
        "Importance": 1,
        "ValueImportanceMap": {
            "IT": 3,
            "HR": 7
        }
    }
]
```

Sorting responses

Amazon Kendra uses the sorting attribute or field as part of the criteria for the documents returned by the query. For example, the results returned by a query sorted by "_created_at" might not contain the same results as a query sorted by "_version".

By default, query responses are sorted by the relevance score that Amazon Kendra determines for each result in the response. To change the sort order, make a document attribute sortable and then configure Amazon Kendra to use that attribute to sort responses.

You can sort results on any built-in or custom attribute/field of the following types:

- Date value
- Long value
- String value

You can't sort attributes of the following type:

- String list values

You can sort on one or more document attributes in each query. Queries return 100 results. If there are fewer than 100 documents with the sorting attribute set, documents without a value for the sorting attribute are returned at the end of the results, sorted by relevance to the query.

To sort document results (AWS SDK)

1. To use the UpdateIndex API to make an attribute sortable, set the `Sortable` parameter to `true`. The following JSON example uses `DocumentMetadataConfigurationUpdates` to add an attribute called "Department" to the index and make it sortable.

```
"DocumentMetadataConfigurationUpdates": [
   {
       "Name": "Department",
       "Type": "STRING_VALUE",
       "Search": {
           "Sortable": "true"
       }
   }
]
```

2. To use one sortable attribute in a query, set the `SortingConfiguration` parameter of the [Query](#) API. Specify the name of the attribute to sort and whether to sort the response in ascending or descending order.

   The following JSON example shows the `SortingConfiguration` parameter that you use to sort the results of a query by the "Department" attribute in ascending order.

   ```
   "SortingConfiguration": {
      "DocumentAttributeKey": "Department",
      "SortOrder": "ASC"
   }
   ```

3. To use more than one sortable attribute in a query, set the `SortingConfigurations` parameter of the [Query](#) API. You can set upto 3 fields that Amazon Kendra should sort the results on. You can also specify whether the results should be sorted in ascending or descending order. The sort field quota can be increased.

   If you don't provide a sorting configuration, the results are sorted by the relevance that Amazon Kendra determines for the result. In the case of ties in sorting the results, the results are sorted by relevance.

   The following JSON example shows the `SortingConfigurations` parameter that you use to sort the results of a query by the attributes "Name" and "Price" in ascending order.

   ```
   "CollapseConfiguration" : {
       "DocumentAttributeKey": "Name",
       "SortingConfigurations": [
           {
               "DocumentAttributeKey": "Price",
               "SortOrder": "ASC"
           }
       ],
       "MissingAttributeKeyStrategy": "IGNORE"
   }
   ```

To sort document results (console)

> **Note**
>
> Multi-attribute sort isn't currently supported by the AWS Management Console.

1. To make an attribute sortable in the console, choose **Sortable** in the attribute definition. You can make an attribute sortable when you create the attribute, or you can modify it later.
2. To sort a query response in the console, choose the attribute to sort the response from the **Sort** menu. Only attributes that were marked sortable during datasource configuration appear in the list.

Collapsing/expanding query results

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

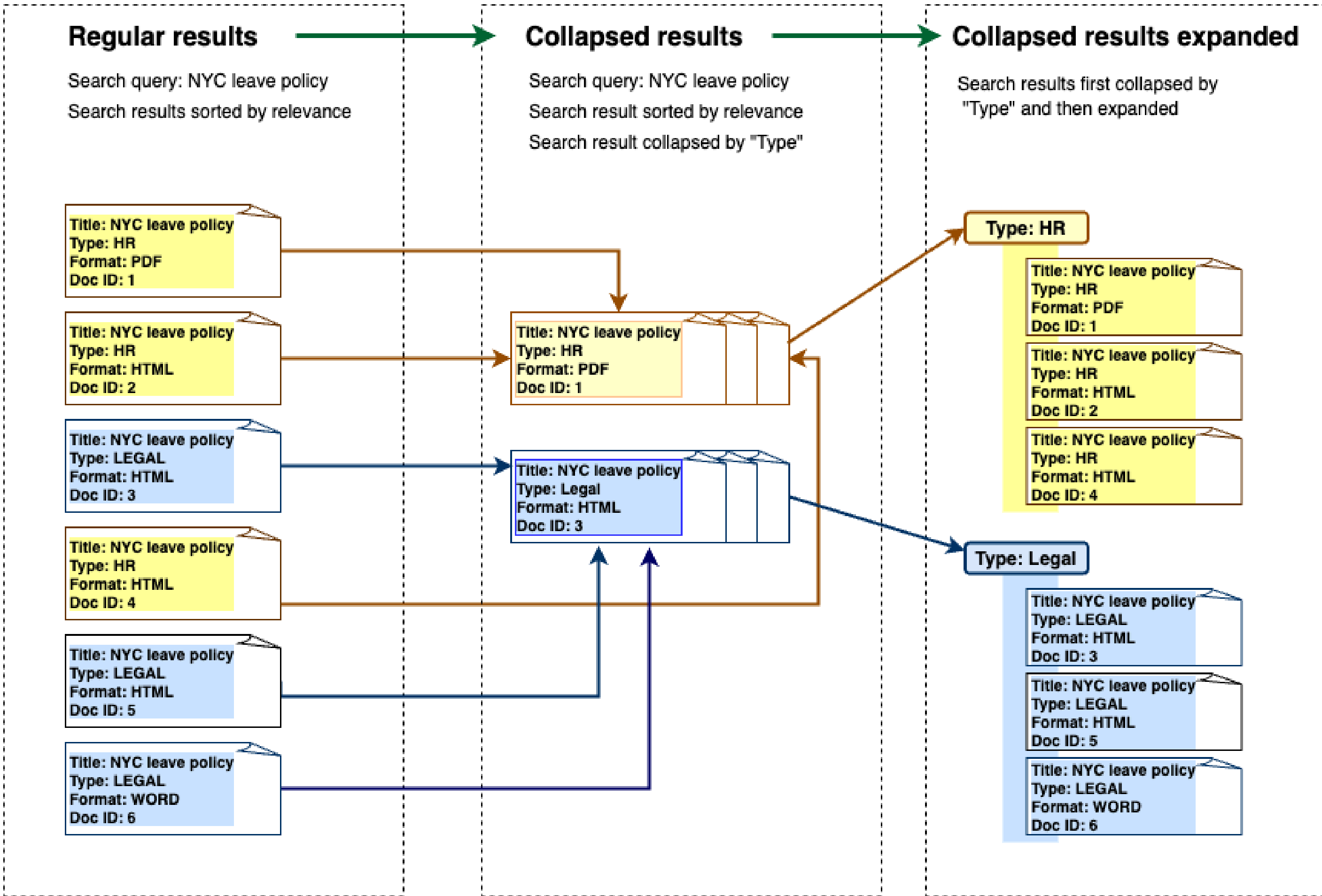
When you connect Amazon Kendra to your data, it crawls document metadata attributes—like `_document_title`, `_created_at`, and `_document_id`—and uses these attributes or fields to provide advanced search capabilities during query time.

Amazon Kendra's Collapse and expand query results feature allows you to group search results using a common document attribute and display them—either collapsed or partially expanded—under a designated primary document.

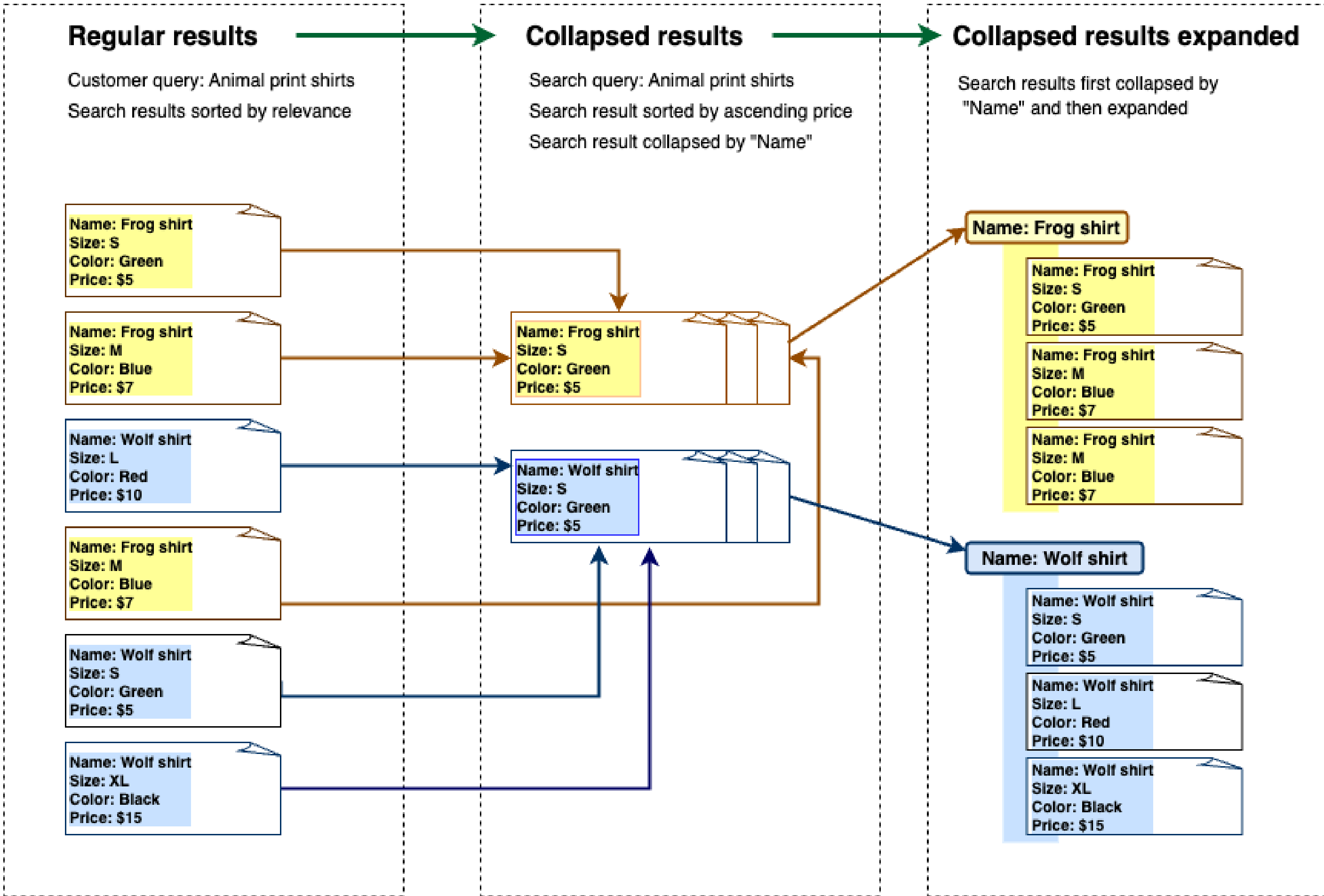
> **Note**
>
> The collapse and expand query results feature is currently available only via the Amazon Kendra API.

This is useful in the following kinds of search situations:

- Multiple versions of content exist in documents within your index. When your end user queries the index, you want them to see the most relevant version of the document with duplicates hidden/collapsed. For example, if your index contains multiple versions of a document named "NYC leave policy" you can choose to collapse the documents for the specific groups "HR" and "Legal" using the "Type" attribute/field.



- Your index contains multiple documents with unique information about one kind of item or object, like a product inventory, for example. To capture and sort item information conveniently, you want end users to access all documents linked by an item or object as one search result. In the example below, a customer search on "animal print shirts" returns results grouped by name, and sorted by ascending price order.



Collapsing results

To group similar or related documents together, you must specify the attribute to collapse on (for example, you can collapse/group documents by `_category`). To do this, call the Query API and use the CollapseConfiguration object to specify the `DocumentAttributeKey` to collapse on. The `DocumentAttributeKey` controls which field search results will be collapsed upon. Supported attribute key fields include `String` and `Number`. `String list` and `Date` type are not supported.

Choosing a primary document using sort order

To configure the primary document to display for a collapsed group, you use the `SortingConfigurations` parameter under CollapseConfiguration. For example, to get the most recent version of a document, you would sort each collapsed group by `_version`. You can specify up to 3 attributes/fields to sort by and a sort order for each attribute/field using `SortingConfigurations`. You can request a quota increase for the number of sort attributes.

By default, Amazon Kendra sorts query responses by the relevance score that it determines for each result in the response. To change the default sort order, make document attributes sortable and then configure Amazon Kendra to use these attributes to sort responses. For more information, see Sorting responses.

Missing document key strategy

If your document doesn't have a collapse attribute value, Amazon Kendra offers three customization options:

- Choose to `COLLAPSE` all documents with null or missing values in one group. This is the default configuration.
- Choose to `IGNORE` documents with null or missing values. Ignored documents will not appear in query results.
- Choose to `EXPAND` each document with a null or missing value into a group of its own.

Expanding results

You can choose whether collapsed search result groups expand using the `Expand` parameter in the CollapseConfiguration object. Expanded results maintain the same sort order that was used to select the primary document for the group.

To configure the number of collapsed search result groups to expand, you use the `MaxResultItemstoExpand` parameter in the ExpandConfiguration object. If you set this value to 10, for example, only the first 10 out of 100 result groups will have expand functionality.

To configure the number of expanded results to show per collapsed primary document, use the `MaxExpandResultsPerItem` parameter. For instance, if you set this value to 3, then at most 3 results per collapsed group will be displayed.

Interactions with other Amazon Kendra features

- Collapsing and expanding results doesn't change the number of facets or impact the total number of results displayed.
- Amazon Kendra featured search results won't be collapsed even if they have the same field value as the collapse field you configure.
- Collapsing and expanding results only applies to results of type `DOCUMENT`.

Tuning search relevance

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see [Index types](https://docs.aws.amazon.com/kendra/latest/dg/hiw-index-types.html).

Amazon Kendra queries produce search results ranked by their relevance. The searchable fields or attributes in the index all contribute to this ranking.

You can modify the effect of a field or attribute on the search relevance through *relevance tuning*. Tuning search relevance can either be done manually at the index level, where you set tuning configurations for your index, or at the query level by overriding configurations set at the index level.

When you use relevance tuning, a result is given a boost in the response when the query includes terms that match the field or attribute. You also specify how much of a boost the document receives when there is a match. Relevance tuning doesn't cause Amazon Kendra to include a document in the query response, it is only one of the factors that Amazon Kendra uses to determine the relevance of a document.

You can boost specific fields or attributes in your index to assign more importance to specific responses. For example when someone searches for "When is re:Invent?" you could boost the relevance of document freshness in the `_last_update_at` field. Or, in an index of research reports, you could boost a specific data source in the "source" field.

You can also boost documents based on votes or view counts which is common in forums and other support knowledge bases. You can combine boosts, for example to boost documents that are viewed more as well as more recent.

You set the amount of boost that a document receives by using the `Importance` parameter. The higher the `Importance`, the more the field or attribute boosts the relevance of a document. When you tune your index or tune at the query level, increase the value of the `Importance` parameter in small increments until you get the effect that you want. To determine if you are improving search results, perform the search and compare the results to previous queries .

You can specify date, number, or string attributes to tune an index or tune at the query level. You can tune fields or attributes that are of the type `StringList` only at the index level. Each field or attribute has specific criteria for when it boosts a result.

- **Date fields or attributes**—There are three specific criteria for date fields, `Duration`, `Freshness` and `RankOrder`.
  - `Duration` sets the time period that the boost applies to. For example, if you set the time period to 86400 seconds (i.e. one day), the boost begins to lessen after one day. The higher the importance, the faster the boost effect lessens.
  - `Freshness` determines how recent a document is when applied to a field or attribute. If you apply `Freshness` to either the field for date created or date last updated, then a more recently created or last updated document is considered "fresher" than an older document. For example, if document 1 was created on November 14, and document 2 was created on November 5, document 1 is "fresher" than document 2. And if document 1 was last updated on November 14, and document 2 was last updated on November 20, document 2 is "fresher" than document 1. The fresher the document, the more this boost is applied. You can only have one `Freshness` field in your index.
  - `RankOrder` applies the boost in either ascending or descending order. If you specify `ASCENDING`, later dates have precedence . If you specify `DESCENDING`, earlier dates have precedence.
- **Number fields or attributes**—For number fields or attributes, you can specify the rank order that Amazon Kendra should use when determining the relevance of the field or attribute. If you specify `ASCENDING`, then higher numbers are given precedence. If you specify `DESCENDING`, then lower numbers have precedence.
- **String fields or attributes**—For string fields or attributes, you can create categories of a field to give each category a different boost. For example, if you boost a field or attribute called "Department", you can give a different boost to documents from "HR" than to documents from "Legal". You can boost a field or attribute of the type `String`. You can boost `StringList` fields only at the index level.

Relevance tuning at the index level

You tune the relevance of a field or attribute at the index level by using either the [console](#) to set tuning in the index details or the [UpdateIndex](#) API.

The following example sets the `_last_updated_at` field as the `Freshness` field for a document.

```
"DocumentMetadataConfigurationUpdates" : [
    {
        "Name": "_last_updated_at",
        "Type": "DATE_VALUE",
        "Relevance": {
            "Freshness": TRUE,
            "Importance": 2
        }
    }
]
```

The following example applies different importance to the different categories in the "department" field.

```
"DocumentMetadataConfigurationUpdates" : [
    {
        "Name": "department",
        "Type": "STRING_VALUE",
        "Relevance": {
            "Importance": 2,
            "ValueImportanceMap": {
                "HR": 3,
                "Legal": 1
            }
        }
    }
]
```

Relevance tuning at the query level

You tune the relevance of a field or attribute at the query level by using the [Query](#) API.

Relevance tuning at the query level is not supported in the console.

Tuning at the query level can speed up the process of testing relevance tuning because you don't need to manually update the tuning configurations in the index for each test. You can tune the relevance of a document by passing tuning configurations in the query. Then you can see the different results that you get from different configurations. A configuration that is passed in the query overrides the configuration that is set at the index level.

The following example overrides the importance applied to the "department" field and each department category set at the index level, shown in the above example. When a user inputs their search query, the "department" field has a fair level of importance and the Legal department has more importance than the HR department.

```
"DocumentRelevanceOverrideConfigurations" : [
    {
        "Name": "department",
        "Type": "STRING_VALUE",
        "Relevance": {
            "Importance": 2,
            "ValueImportanceMap": {
                "HR": 2,
                "Legal": 8
            }
        }
    }
]
```

Gaining insights with search analytics

> **Note**
>
> Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

You can use Amazon Kendra search *Analytics* to gain insights on how your search application is successfully or unsuccessfully helping your users find information.

Amazon Kendra Analytics provide a snapshot of how your users interact with your search application and how effective your search application configuration is. You can view the metrics data using the GetSnapshots API or by selecting **Analytics** on the navigation panel in the console.

You can render the data generated by `GetSnapshots` on your own custom-built dashboard. Or you can use the metrics dashboard provided in the console, which includes visual graphs. With a visual dashboard, you can look for trends or patterns in user behavior over time or surface problems with your search application configuration. For example, a line graph that shows a consistent number of queries per day and a steady increase might indicate increased adoption and usage. On the other hand, an abrupt drop might indicate there's an issue that must be investigated.

You can use the metrics to make connections between different points of data to solve problems with how your users query for information or discover business opportunities. For example, the document 'How does AI work?' is the most clicked on document in the search results, and the top searched query is 'How does machine learning work?'. This informs you on the preferred terms and language your users use. You can integrate these terms in your documents or use custom synonyms for these terms to make your documents more searchable for your users.

Metrics for search

There are 10 metrics for analyzing your search application's performance or what information your users are searching for. To retrieve the metrics data, you specify the string name of the metric data you want to retrieve when you call `GetSnapshots`.

You also must provide a time interval or time window to view the metrics data. The time interval uses the time zone of your index. You can view data in the following time windows:

- THIS_WEEK: The current week, starting on the Sunday and ending on the day before the current date.
- ONE_WEEK_AGO: The previous week, starting on the Sunday and ending on the following Saturday.
- TWO_WEEKS_AGO: The week before the previous week, starting on the Sunday and ending on the following Saturday.
- THIS_MONTH: The current month, starting on the first day of the month and ending on the day before the current date.
- ONE_MONTH_AGO: The previous month, starting on the first day of the month and ending on the last day of the month.
- TWO_MONTHS_AGO: The month before the previous month, starting on the first day of the month and ending on last day of the month.

In the console, the supported time windows are **This week**, **Previous week**, **This month**, **Previous month**.

Click-through rate

The proportion of queries that lead to click-through to a document in the search results. This helps you understand if your search application configuration helps your users find information relevant to their queries. For queries that return instant answers, users might not need to click through to a document for more information. For more information, see the section called "Instant answer rate". You must call SubmitFeedback to ensure that click-through feedback is collected.

To retrieve data on click-through rate using the GetSnapshots API, specify the metricType as AGG_QUERY_DOC_METRICS. You can also view this metric in the console by selecting **Analytics** on the navigation panel.

Zero click rate

The proportion of queries that lead to zero clicks in the search results. This helps you understand gaps in your content providing irrelevant search results. For queries that return instant answers, users might not need to click through to a document for more information. For more information, see the section called "Instant answer rate". Also, your search settings, such as tuning configurations, could have an impact on how documents are returned in the search results.

To retrieve data on zero click rate using the `GetSnapshots` API, specify the `metricType` as `AGG_QUERY_DOC_METRICS`. You can also view this metric in the console by selecting **Analytics** on the navigation panel.

Zero search results rate

The proportion of queries that lead to zero search results. This helps you understand gaps in your content providing no relevant search results.

To retrieve data on zero search results rate using the `GetSnapshots` API, specify the `metricType` as `AGG_QUERY_DOC_METRICS`. You can also view this metric in the console by selecting **Analytics** on the navigation panel.

Instant answer rate

The proportion of queries with an instant answer or FAQ returned. This helps you understand the role of instant answers in providing information.

To retrieve data on instant answer rate using the `GetSnapshots` API, specify the `metricType` as `AGG_QUERY_DOC_METRICS`. You can also view this metric in the console by selecting **Analytics** on the navigation panel.

Top queries

The top 100 queries searched by your users. This helps you understand which queries are popular and the kind of information your users are most interested in.

Metrics include the number of times the query is searched, the proportion of click-throughs to a document, the proportion of no click-throughs to a document, the average click depth in the search results for the query, the proportion of instant answers for the query, and the average confidence for the first 10 search results for a query.

To retrieve data on top queries using the `GetSnapshots` API, specify the `metricType` as `QUERIES_BY_COUNT`. You can also view this metric in the console by selecting **Analytics** on the navigation panel in the console, then selecting **Top queries** under **Query lists**.

Top queries with zero clicks

The top 100 queries that lead to zero clicks in the search results. This helps you understand any gaps in your content, where there's a lack of documents relevant to some queries or your search application configuration is returning irrelevant search results. For queries that return instant answers, users might not need to click through to a document for more information. For more information, see the section called “Instant answer rate”.

Metrics include the number of times the query leads to zero clicks, the proportion of zero clicks for the query, the proportion of instant answers for the query, and the average confidence for the first 10 search results for a query.

To retrieve data on top queries with zero clicks using the `GetSnapshots` API, specify the `metricType` as `QUERIES_BY_ZERO_CLICK_RATE`. You can also view this metric in the console by selecting **Analytics** on the navigation panel in the console, then selecting **Top zero click queries** under **Query lists**.

Top queries with zero search results

The top 100 queries that lead to zero search results. This helps you understand any gaps in your content, where there are no documents relevant to some queries. Or, your users might query with specialized terms that possibly lead to no search results, prompting you to create custom synonyms to handle this.

Metrics include the number of times the query leads to zero search results, the proportion of zero search results for the query, and the proportion of times the query is searched compared to all queries.

To retrieve data on top queries with zero search results using the `GetSnapshots` API, specify the `metricType` as `QUERIES_BY_ZERO_RESULT_RATE`. You can also view this metric in the console by selecting **Analytics** on the navigation panel in the console, then selecting **Top zero result queries** under **Query lists**.

Top clicked on documents

The top 100 most clicked on documents in the search results. This helps you understand which documents or search results are most relevant to your users when they query for information.

Metrics include the number of times the document is clicked on, the number of likes a document receives from your users (thumbs up), the number of dislikes a document receives from your users (thumbs down).

To retrieve data on top clicked on documents using the `GetSnapshots` API, specify the `metricType` as `DOCS_BY_CLICK_COUNT`. You can also view this metric in the console by selecting

Analytics on the navigation panel in the console, then selecting **Top clicked documents** under **Query lists**.

Total queries

The total number of queries searched by your users. This helps you understand how engaged your users are with your search application.

To retrieve data on total queries using the `GetSnapshots` API, specify the `metricType` as `AGG_QUERY_DOC_METRICS`. You can also view this metric in the console by selecting **Analytics** on the navigation panel.

Total documents

The total number of documents in your index. This helps you compare the size of your index to the total number of queries to check if there is an appropriate number of documents for the volume of queries.

To retrieve data on total documents using the `GetSnapshots` API, specify the `metricType` as `AGG_QUERY_DOC_METRICS`. You can also view this metric in the console by selecting **Analytics** on the navigation panel.

Example of retrieving metric data

The following code is an example of retrieving data on the top queries for the previous month.

Console

To retrieve top queries for the previous month

1. In the left navigation pane, under **Indexes**, select your index, and then select **Analytics**.
2. On the **Analytics** page, select the button **This week**, to change the time window for retrieving the data to **Previous month**.
3. On the **Analytics** page, under **Query lists**, select **Top queries**.

CLI

To retrieve top queries for the previous month

```
aws kendra get-snapshots \
--index-id index-id \
--interval "ONE_MONTH_AGO" \
--metric-type "QUERIES_BY_COUNT"
```

Python

To retrieve top queries for the previous month

```
import boto3

kendra = boto3.client("kendra")

index_id = "index-id"
interval = "ONE_MONTH_AGO"
metric_type = "QUERIES_BY_COUNT"

snapshots_response = kendra.get_snapshots(
    IndexId = index_id,
    Interval = interval,
    MetricType = metric_type
)

print("Top queries data: " + snapshots_response["snapshotsData"])
```

Java

To retrieve top queries for the previous month

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.GetSnapshotsRequest;
import software.amazon.awssdk.services.kendra.model.GetSnapshotsResponse;

public class TopQueriesExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        String indexId = "indexID";
        String interval = "ONE_MONTH_AGO";
        String metricType = "QUERIES_BY_COUNT";
```

```
        GetSnapshotsRequest getSnapshotsRequest = GetSnapshotsRequest
            .builder()
            .indexId(indexId)
            .interval(interval)
            .metricType(metricType)
            .build();

        GetSnapshotsResponse getSnapshotsResponse =
kendra.getSnapshots(GetSnapshotsRequest);

        System.out.println(String.format("Top queries data: ",
getSnapshotsResponse.snapshotsData()))
```

From metrics to actionable insights

Actionable insights are meaningful pieces of information extracted from raw data and are used to guide your actions or decisions. To extract meaning from the metrics and use them to drive actionable insights, it is important to not only look at the metrics in isolation but also make connections among the metrics.

For example, the top query with zero clicks is 'Which regions are currently available?'. However, it also has a 100 percent instant answer rate. This suggests your users receive the answer to this question without needing to click on a search result or document that provides information on available regions. If you looked at zero clicks alone, you would not get the full story and possibly make the wrong conclusions about the success of your search application configuration in handling this query.

Another example of an actionable insight is discovering a business opportunity. Businesses often look for opportunities to grow their customers by analyzing search metrics. The most clicked on document is 'Available regions'. In addition to this, most of the top searched queries are related to questions on product availability in the Oceanic region, with 100 percent instant answer rates and a high click-through rate to more information on available regions as part of the answer. This suggests there's interest and demand for your product or service in this region.

Visualizing and reporting search analytics

There are five metrics that include trends data for you to visualize and look for trends or patterns over time. If you use the console, graphs of the trends data are provided. If you use the APIs, you can retrieve the trends data to create your own graphs or visualizations. Most graphs in the console plot the daily data points over your chosen time window.

The console provides a dashboard of the metrics where you can select a graph and top list you are interested in viewing. You can export the metrics shown on your dashboard in CSV format by selecting **Export** on the **Analytics** home page. You can include these reports in your business documents or presentations.

You can visualize the following metrics:

Total queries graph

A line graph of the number of queries issued per day. The graph helps you visualize patterns in daily user engagement. Some examples include a steady increase or decrease in user engagement, or a drastic drop to 0 queries due to a crash of your search application or issues with your website.

If you use the API, you can retrieve these data by specifying `TREND_QUERY_DOC_METRICS`. You can use the data to create your own graphs, or use the graphs provided in the console.

Click-through rate graph

A line graph of the proportions of click-throughs per day. The graph helps you visualize patterns in daily click-through rate. Some examples include a steady increase or decrease in click-through rate, or a decrease in instant answers possibly influencing an increase in click-through.

If you use the API, you can retrieve these data by specifying `TREND_QUERY_DOC_METRICS`. You can use the data to create your own graphs, or use the graphs provided in the console.

Zero click rate graph

A line graph of the proportion of zero clicks per day. The graph helps you visualize patterns in daily zero click rate. Some examples include a steady increase or decrease in zero click rate, or an increase in instant answers possibly influencing an increase in zero clicks.

If you use the API, you can retrieve these data by specifying `TREND_QUERY_DOC_METRICS`. You can use the data to create your own graphs, or use the graphs provided in the console.

Zero search results rate graph

A line graph of the proportion of zero search results per day. The graph helps you visualize patterns in daily zero search results rate. Some examples include a steady increase or decrease in zero search results rate, or a sharp decrease in the number of documents in your index possibly influencing an increase in zero search results.

If you use the API, you can retrieve these data by specifying `TREND_QUERY_DOC_METRICS`. You can use the data to create your own graphs, or use the graphs provided in the console.

Instant answer rate graph

A line graph of the proportion of queries with an instant answer or FAQ returned. The graph helps you visualize patterns in daily instant answer rate. Some examples include steady increase or decrease in question-answer type queries, or a decrease in click-throughs possibly influencing an increase in instant answers.

If you use the API, you can retrieve these data by specifying `TREND_QUERY_DOC_METRICS`. You can use the data to create your own graphs, or use the graphs provided in the console.

Submitting feedback for incremental learning

Note

Feature support varies by index type and search API being used. To see if this feature is supported for the index type and search API you're using, see Index types.

Amazon Kendra uses incremental learning to improve search results. Using feedback from queries, incremental learning improves the ranking algorithms and optimizes search results for greater accuracy.

For example, suppose that your users search for the phrase "health care benefits." If users consistently choose the second result from the list, over time Amazon Kendra boosts that result to the first place result. The boost decreases over time, so if users stop selecting a result, Amazon Kendra eventually removes it and shows another more popular result instead. This helps Amazon Kendra prioritize results based on relevance, age, and content.

Incremental learning is activated for all indexes and for all supported document types.

Amazon Kendra starts learning as soon as you provide feedback, though it can take over 24 hours to see the results of the feedback. Amazon Kendra provides three methods for you to submit feedback: the AWS console, a JavaScript library that you can include on your search results page, and an API that you can use.

Amazon Kendra accepts two types of user feedback:

- **Clicks**—Information about which query results the user chose. The feedback includes the result ID and the Unix timestamp of the date and time that the search result was chosen.

  To submit click feedback, your application must collect click information from the activities of your users, and then submit that information to Amazon Kendra. You can collect click information with the console, the JavaScript library, and the Amazon Kendra API.
- **Relevance**—Information about the relevance of a search result, which the user typically provides. The feedback contains the result ID and a relevance indicator (`RELEVANT` or `NOT_RELEVANT`). The user determines the relevance information.

  To submit relevance feedback, your application must provide a feedback mechanism that allows the user to choose the appropriate relevance for a query result, and then submit that information to Amazon Kendra. You can only collect relevance information with the console and the Amazon Kendra API.

Feedback is used while the index is active. Feedback only affects the index that it is submitted to, it can't be used across indexes or for different accounts.

You should provide additional user context when you query your Amazon Kendra index. When you provide user context, Amazon Kendra is able to tell if the feedback is provided by a single user or by multiple users and adjust search results accordingly.

When you provide user context, the feedback for the query is associated with the specific user provided in the context. If you don't specify user context, you can provide a visitor ID that is used to group and aggregate queries.

If you don't provide user context or a visitor ID, the feedback is anonymous and aggregated with other anonymous feedback.

The following code shows how to include user context as a token or the visitor ID.

```
response = kendra.query(
    QueryText = query,
    IndexId = index,
    UserToken = {
        Token = "token"
    })

    OR

    response = kendra.query(
    QueryText = query,
    IndexId = index,
    VisitorId = "visitor-id")
```

For web applications, you can use cookies, locations, or browser users to generate a visitor ID for each user.

For head queries, the largest volume of queries, providing click-through feedback provides enough information to improve overall accuracy. For tail queries, those that are rare, subject matter experts should submit relevant and non-relevant feedback to improve accuracy for those queries.

In addition to the console, you can use one of two methods: a JavaScript library or the SubmitFeedback API. You should only use one method of gathering feedback. For best results, you should submit feedback within 24 hours of making the query.

Topics

Using the Amazon Kendra JavaScript library to submit feedback

Amazon Kendra provides a JavaScript library that you can use to add click feedback to your search results page. To use the library, you insert a script tag in your client code that displays the search result, then add information to each of the document links in your result list. When a user chooses a link to view a document, click information is sent to Amazon Kendra.

The library works with browsers that support JavaScript version ES6/ES2015.

Step 1: Insert a script tag into your Amazon Kendra search application

In your client code that renders the Amazon Kendra search results, insert a <script> tag and add a reference to the JavaScript library:

```
<script>
 (function(w, d, s, c, g, n) {
   if(!w[n]) {
     w[n] = w[n] || function () {
           (w[n].q = w[n].q || []).push(arguments);
     }
     w[n].st = new Date().getTime();
     w[n].ep = g;
     var e = document.createElement(s),
         j = document.getElementsByTagName(s)[0];
     e.async = 1;
     e.src = c;
     e.type = 'module';
     j.parentNode.insertBefore(e, j);
   }
 })(window, document, 'script',
```

```
 'library download URL',
 'feedback endpoint',
 'kendraFeedback');
</script>
```

The script asynchronously downloads the JavaScript library from an Amazon Kendra hosted CDN and initializes a global variable called `kendraFeedback` that allows you to set optional parameters.

Replace *`library download URL`* and *`feedback endpoint`* with an identifier from the following table based on the region that hosts your Amazon Kendra index.

Region	Download URL	Feedback endpoint
us-east-1	https://d2zm0lpns956f8.cloudfront.net/ksf-v1.js	https://ujxwp5s92h.execute-api.us-east-1.amazonaws.com/prod/submit
us-east-2	https://d2crv7fufeg244.cloudfront.net/ksf-v1.js	https://i6h76zwzf3.execute-api.us-east-2.amazonaws.com/prod/submit
us-west-2	https://d2iezfpnpcoujy.cloudfront.net/ksf-v1.js	https://wg6nim909c.execute-api.us-west-2.amazonaws.com/prod/submit
ca-central-1	https://d1zbkfomowykaq.cloudfront.net/ksf-v1.js	https://budi8txevj.execute-api.ca-central-1.amazonaws.com/prod/submit
eu-west-1	https://d3gptlxtulu4us.cloudfront.net/ksf-v1.js	https://po2b11740b.execute-api.eu-west-1.amazonaws.com/prod/submit
ap-southeast-1	https://d1vvuam7g4taoe.cloudfront.net/ksf-v1.js	https://9je5uw7t5l.execute-api.ap-southeast-1.amazonaws.com/prod/submit

Region	Download URL	Feedback endpoint
ap-southeast-2	https://dopqntoe6z0ce.cloudfront.net/ksf-v1.js	https://oovf4nvjj7.execute-api.ap-southeast-2.amazonaws.com/prod/submit
ap-south-1	https://d1ts9ouelsmk3g.cloudfront.net/ksf-v1.js	https://k1abnmd43b.execute-api.ap-south-1.amazonaws.com/prod/submit
ap-northeast-1	https://d3w0ybsa293kb4.cloudfront.net/ksf-v1.js	https://wg7rz0uzjh.execute-api.ap-northeast-1.amazonaws.com/prod/submit
eu-west-2	https://d1tsrujswld1d1.cloudfront.net/ksf-v1.js	https://qi7mct3x7f.execute-api.eu-west-2.amazonaws.com/prod/submit

For example, if your index is in US East (N. Virginia), *`library download URL`* is `https://d2zm0lpns956f8.cloudfront.net/ksf-v1.js` and *`feedback endpoint`* is `https://ujxwp5s92h.execute-api.us-east-1.amazonaws.com/prod/submit`.

There are two optional settings that you can make for the Amazon Kendra JavaScript library:

- `disableCookies` – By default, Amazon Kendra sets a cookie that uniquely identifies the user. Set this to `true` to disable the cookie.

```
kendraFeedback('disableCookie', 'true | false');
```

  `searchDivClassName` – By default, Amazon Kendra monitors all links on your search results page for clicks. Set this to a `<div>` class name to monitor only links in the specified class.

```
kendraFeedback('searchDivClassName', 'class name');
```

Step 2: Add the feedback token to search results

On your result page, add an HTML attribute called `data-kendra-token` to the anchor tag or immediate parent div tag that contains a link to the document from the query response. For example:

```
<a href="document location" data-kendra-token="feedback token value"></a>
OR
<div data-url="document location" data-kendra-token="feedback token value"></div>
```

A query response contains a token in the `feedbackToken` field. The token uniquely identifies the response if the user chooses it. Assign the value of the token to the `data-kendra-token` attribute. The Amazon Kendra JavaScript library looks for this token when the user chooses the result and submits it to an Amazon Kendra endpoint as feedback.

The Amazon Kendra JavaScript library only submits the feedback token and other metadata such as the time the result was chosen and a unique visitor ID.

Step 3: Test the feedback script

To make sure that the JavaScript library is configured correctly and sending feedback to the right endpoint, do the following. This example uses the Chrome browser.

1. Open the Web developer tools in the browser. On Chrome, open the **Chrome menu** in the upper right corner of the browser, choose **More tools** and then choose **Developer tools**.
2. Make sure that there are no errors related to the Amazon Kendra JavaScript library in the console tab.
3. Make a search and choose any result. In the **Network** tab of the developer tools. You should see a request sent to the feedback endpoint, the token for the result, and a 200 OK status.

Using the Amazon Kendra API to submit feedback

To use the Amazon Kendra API to submit query feedback, use the [SubmitFeedback](#) API. To identify the query, you supply the index ID of the index that the query applies to, and the query ID returned in the response from the [Query](#) API.

The following example shows how to submit click and relevance feedback using the Amazon Kendra API. You can submit multiple sets of feedback through the `ClickFeedbackItems` and

`RelevanceFeedbackItems` arrays. This example submits a single click and a single relevance feedback item. The feedback submittal uses the current time.

To submit feedback for a search (AWS SDK)

1. You can use the following example code with the required values:

   a. `index id`—The ID of the index that the query applies to.

   b. `query id`—The query that you want to provide feedback on.

   c. `result id`—The ID of the query result that you want to provide feedback on. The query response contains the result ID.

   d. `relevance value`—Either `RELEVANT` (the query result is relevant) or `NOT_RELEVANT` (the query result is not relevant).

Python

```
import boto3
import time

kendra = boto3.client("kendra")

# Provide the index ID
index_id = "index-id"
# Provide the query ID
query_id = "query-id"
# Provide the search result ID
result_id = "result-id"

# Configure the feedback item
feedback_item = {"ClickTime": int(time.time()),
    "ResultId":result_id}

# Configure the relevance value
relevance_value = "RELEVANT"
relevance_item = {"RelevanceValue": relevance_value,
    "ResultId": result_id
    }

response = kendra.submit_feedback(
    QueryId = query_id,
    IndexId = index_id,
```

```
    ClickFeedbackItems = [feedback_item],
    RelevanceFeedbackItems = [relevance_item]
)


print("Submitted feedback for query: " + query_id)
```

Java

```
package com.amazonaws.kendra;

import java.time.Instant;
import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.ClickFeedback;
import software.amazon.awssdk.services.kendra.model.RelevanceFeedback;
import software.amazon.awssdk.services.kendra.model.RelevanceType;
import software.amazon.awssdk.services.kendra.model.SubmitFeedbackRequest;
import software.amazon.awssdk.services.kendra.model.SubmitFeedbackResponse;

public class SubmitFeedbackExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        SubmitFeedbackRequest submitFeedbackRequest = SubmitFeedbackRequest
            .builder()
            .indexId("IndexId")
            .queryId("QueryId")
            .clickFeedbackItems(
                ClickFeedback
                .builder()
                .clickTime(Instant.now())
                .resultId("ResultId")
                .build())
            .relevanceFeedbackItems(
                RelevanceFeedback
                .builder()
                .relevanceValue(RelevanceType.RELEVANT)
                .resultId("ResultId")
```

```
                .build())
            .build();

        SubmitFeedbackResponse response =
 kendra.submitFeedback(submitFeedbackRequest);

        System.out.println("Feedback is submitted");
    }
}
```

2. Run the code. After the feedback has been submitted, the code displays a message.

Adding custom synonyms to an index

To add custom synonyms to an index, you specify them in a thesaurus file. You can include business-specific or specialized terms in Amazon Kendra using synonyms. Generic English synonyms, such as `leader, head`, are built into Amazon Kendra and should not be included in a thesaurus file, including generic synonyms that use hyphens. Amazon Kendra supports synonyms for all response types, which include `DOCUMENT` response types and `QUESTION_ANSWER` or `ANSWER` response types. Amazon Kendra currently does not support adding synonyms flagged as stopwords. This is to be included in a future release.

Amazon Kendra makes correlations between synonyms. For example, using the synonym pair `Dynamo, Amazon DynamoDB`, Amazon Kendra correlates Dynamo with Amazon DynamoDB. The query "What is dynamo?" then returns a document such as "What is Amazon DynamoDB?". With synonyms, Amazon Kendra can more easily pick up the correlation.

The thesaurus file is a text file stored in an Amazon S3 bucket. See Adding a thesaurus to an index.

The thesaurus file uses the Solr synonym format. Amazon Kendra has a limit on the number of thesauri per index. See Quotas.

Synonyms can be useful in the following scenarios:

- Specialized terms that are not traditional English language synonyms such as `NLP, Natural Language Processing`.
- Proper nouns with complex semantic associations. These are nouns that the general public are unlikely to understand, for example, in machine learning, `cost, loss, model performance`.
- Different forms of product names, for example, `Elastic Compute Cloud, EC2`.
- Domain-specific or business-specific terms, such as product names. For example, `Route53, DNS`.

Do not use synonyms in the following scenarios:

- Generic English language synonyms such as `leader, head`. These synonyms are not domain-specific,and using synonyms in these scenarios might have unintended effects.
- Typographical errors such as `teh => the`.

- Morphological variants like the plurals and possessives of nouns, the comparative and superlative form of adjectives, and the past tense, past participle and progressive form of verbs. One example of comparative and superlative adjectives is `good, better, best`.
- Unigram (single word) stop words such as `WHO`. Unigram stop words are not allowed in the thesaurus and are excluded from search. For example, `WHO => World Health Organization` is rejected. You can use `W.H.O.` however as a synonym term, and you can use stop words as part of a multi-word synonym. For example, `of` is not allowed but `United States of America` is accepted.

Custom synonyms make it easy to improve Amazon Kendra's understanding of your business-specific terminology by expanding your queries to cover your business-specific synonyms. Although synonyms can improve search accuracy, it is important to understand how synonyms affect latency so you can optimize for this.

A general rule for synonyms is: the more terms in your query that are matched and expanded with synonyms, the greater potential impact on latency. Other factors that affect latency include the average size of documents indexed, the size of your index, any filtering on search results, and the overall load on your Amazon Kendra index. Queries that don't match any synonyms are not affected.

A general guideline for how synonyms affect latency:

Use case	Increase in latency*
Typical natural language or keyword queries of 3 to 5 words each	Less than 15 percent
1 query term expands to 3 synonyms	
Index of about 500,000 documents (averaging 10.48 KB of extracted text per document) or 30,000 FAQ / question pairs	

Note: the "Less than 15 percent" cell spans all three use-case rows.

**Performance varies based on your specific use of synonyms and configurations on your index. It's best to test search performance to obtain more accurate benchmarks for your specific use case.*

If your thesaurus is large, has a high term expansion ratio, and your latency increase is not within acceptable boundaries, you can try one or both of the following:

- Trim your thesaurus to reduce the expansion ratio (number of synonyms per term).
- Trim the overall coverage of terms (number of lines in your thesaurus).

Alternatively, you can increase the provisioning capacity (virtual storage units) to offset the latency increase.

Topics

Creating a thesaurus file

An Amazon Kendra thesaurus file is a UTF-8-encoded file containing a list of synonyms in the Solr synonym list format. The thesaurus file must be less than 5 MB.

There are two ways to specify synonym mappings:

- *Bidirectional synonyms* are specified as a comma-separated list of terms. If your user queries any of the terms, then all the terms in the list are used to search documents, which includes the original queried term.
- *Unidirectional synonyms* are specified as terms separated by the symbol "=>" between them to map terms to their synonyms. If your user queries a term on the left of the symbol "=>", then it is mapped to a term on the right to search for documents using the synonym. It is not mapped vice versa, making this unidirectional.

The synonyms themselves are case sensitive, but the terms they map to are case insensitive. For example, `ML => Machine Learning` means if your user queries "ML" or "ml" or uses some other case, it will map to "Machine Learning". If you were to map this vice versa, `Machine Learning => ML`, then "Machine Learning" or "machine learning" or some other case would map to "ML".

A synonym doesn't search for an exact match on special characters. For example, if you search for "dead-letter-queue", Amazon Kendra can return documents that match "dead letter queue" (no hyphen). If your documents contain hyphens, such as "dead-letter-queue", Amazon Kendra processes the documents during search to remove hyphens. For generic English synonym terms that are built into Amazon Kendra and should not be included in a thesaurus file, Amazon Kendra can search both the hyphen version of the term and the non-hyphen version of the term. For example, if you search "third-party" and "third party", Amazon Kendra returns documents that match either version of those terms.

For synonyms that contain stopwords or commonly used words, Amazon Kendra returns documents that match terms including stopwords. For example, you can create a synonym rule to map "on boarding" and "onboarding". You cannot use stopwords alone for synonyms. For example, if you search for "on", Amazon Kendra cannot return all documents that contain "on".

Some synonym rules are ignored. For example, `a => b` is a rule, but `a => a` is ignored and doesn't count as a rule.

The term count is the number of unique terms in the theaurus file. The below example file includes terms `AWS CodeStar`, `ML`, `Machine Learning`, `autoscaling group`, `ASG`, and more.

There is a maximum amount of synonym rules per thesaurus and a maximum amount of synonyms per term. For more information, see Quotas for Amazon Kendra.

The following example shows a thesaurus file with synonym rules. Each line contains a single synonym rule. Blank lines and comments are ignored.

```
# Lines starting with pound are comments and blank lines are ignored.

# Synonym relationships can be defined as unidirectional or bidirectional
 relationships.

# Unidirection relationships are represented by any term sequence
# on the left hand side (LHS) of "=>" followed by synonyms on the right hand side (RHS)
CodeStar => AWS CodeStar
# This will map CodeStar to AWS CodeStar, but not vice-versa

# To map terms vice versa
ML => Machine Learning
Machine Learning => ML

# Multiple synonym relationships may be defined in one line as well by comma
 seperation.
```

```
autoscaling group, ASG => Auto Scaling group, autoscaling
# The above is equivalent to:
# autoscaling group => Auto Scaling group, autoscaling
# ASG => Auto Scaling group, autoscaling

# Bi-directional synonyms are comma separated terms with no "=>"
DNS, Route53, Route 53
# DNS, Route53, and Route 53 map to one another and are interchangeable at match time
# The above is equivalent to:
# DNS => Route53, Route 53
# Route53 => DNS, Route 53
# Route 53 => DNS, Route53

# Overlapping LHS terms will be merged
Beta => Alpha
Beta => Gamma
Beta, Delta
# is equivalent to:
# Beta => Alpha, Gamma, Delta
# Delta => Beta

# Each line contains a single synonym rule.
# Synonym rule count is the total number of lines defining synonym relationships
# Term count is the total number of unique terms for all rules.
# Comments and blanks lines do not count.
```

Adding a thesaurus to an index

The following procedures show how to add a thesaurus file containing synonyms to an index. It can take up to 30 minutes to see the effects of your updated thesaurus file. For more information about the thesaurus file, see Creating a thesaurus file.

Console

To add a thesaurus

1. In the left navigation pane, under the index where you want to add a list of synonyms, your thesaurus, choose **Synonyms**.
2. On the **Synonym** page, choose **Add Thesaurus**.
3. In **Define thesaurus**, give your thesaurus a name and an optional description.

4. In **Thesaurus settings**, provide the Amazon S3 path to your thesaurus file. The file must be smaller than 5 MB.
5. For **IAM Role**, select a role or select **Create a new role** and specify a role name to create a new role. Amazon Kendra uses this role to access the Amazon S3 resource on your behalf. The IAM role has the prefix "AmazonKendra-".
6. Choose **Save** to save the configuration and add the thesaurus. Once the thesaurus is ingested, it is active and synonyms are highlighted in results. It can take up to 30 minutes to see the effects of your thesaurus file.

CLI

To add a thesarus to an index with the AWS CLI, call `create-thesaurus`:

```
aws kendra create-thesaurus \
--index-id index-id \
--name "thesaurus-name" \
--description "thesaurus-description" \
--source-s3-path "Bucket=bucket-name,Key=thesaurus/synonyms.txt" \
--role-arn role-arn
```

Call `list-thesauri` to see a list of thesauruses:

```
aws kendra list-thesauri \
--index-id index-id
```

To view details for a thesaurus, call `describe-thesaurus`:

```
aws kendra describe-thesaurus \
--index-id index-id \
--index-id thesaurus-id
```

It can take up to 30 minutes to see the effects of your thesaurus file.

Python

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time
```

```
kendra = boto3.client("kendra")

print("Create a thesaurus")

thesaurus_name = "thesaurus-name"
thesaurus_description = "thesaurus-description"
thesaurus_role_arn = "role-arn"

index_id = "index-id"

s3_bucket_name = "bucket-name"
s3_key = "thesaurus-file"
source_s3_path= {
    'Bucket': s3_bucket_name,
    'Key': s3_key
}

try:
    thesaurus_response = kendra.create_thesaurus(
        Description = thesaurus_description,
        Name = thesaurus_name,
        RoleArn = thesaurus_role_arn,
        IndexId = index_id,
        SourceS3Path = source_s3_path
    )

    pprint.pprint(thesaurus_response)

    thesaurus_id = thesaurus_response["Id"]

    print("Wait for Kendra to create the thesaurus.")

    while True:
        # Get thesaurus description
        thesaurus_description = kendra.describe_thesaurus(
            Id = thesaurus_id,
            IndexId = index_id
        )
        # If status is not CREATING quit
        status = thesaurus_description["Status"]
        print("Creating thesaurus. Status: " + status)
        if status != "CREATING":
            break
        time.sleep(60)
```

```
except ClientError as e:
        print("%s" % e)

print("Program ends.")
```

Java

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.CreateThesaurusRequest;
import software.amazon.awssdk.services.kendra.model.CreateThesaurusResponse;
import software.amazon.awssdk.services.kendra.model.DescribeThesaurusRequest;
import software.amazon.awssdk.services.kendra.model.DescribeThesaurusResponse;
import software.amazon.awssdk.services.kendra.model.S3Path;
import software.amazon.awssdk.services.kendra.model.ThesaurusStatus;

public class CreateThesaurusExample {

  public static void main(String[] args) throws InterruptedException {

    KendraClient kendra = KendraClient.builder().build();

    String thesaurusName = "thesaurus-name";
    String thesaurusDescription = "thesaurus-description";
    String thesaurusRoleArn = "role-arn";

    String s3BucketName = "bucket-name";
    String s3Key = "thesaurus-file";
    String indexId = "index-id";

    System.out.println(String.format("Creating a thesaurus named %s",
 thesaurusName));
    CreateThesaurusRequest createThesaurusRequest = CreateThesaurusRequest
        .builder()
        .name(thesaurusName)
        .indexId(indexId)
        .description(thesaurusDescription)
        .roleArn(thesaurusRoleArn)
        .sourceS3Path(S3Path.builder()
            .bucket(s3BucketName)
            .key(s3Key)
```

```
            .build())
        .build();
    CreateThesaurusResponse createThesaurusResponse =
 kendra.createThesaurus(createThesaurusRequest);
    System.out.println(String.format("Thesaurus response %s",
 createThesaurusResponse));

    String thesaurusId = createThesaurusResponse.id();

    System.out.println(String.format("Waiting until the thesaurus with ID %s is
 created.", thesaurusId));

    while (true) {
      DescribeThesaurusRequest describeThesaurusRequest =
 DescribeThesaurusRequest.builder()
          .id(thesaurusId)
          .indexId(indexId)
          .build();
      DescribeThesaurusResponse describeThesaurusResponse =
 kendra.describeThesaurus(describeThesaurusRequest);
      ThesaurusStatus status = describeThesaurusResponse.status();
      if (status != ThesaurusStatus.CREATING) {
        break;
      }

      TimeUnit.SECONDS.sleep(60);
    }

    System.out.println("Thesaurus creation is complete.");
  }
}
```

Updating a thesaurus

You can change the configuration of a thesaurus after it is created. You can change details like thesaurus name and IAM information. You can also change the location of the thesaurus file Amazon S3 path. If you change the path to the thesaurus file, Amazon Kendra replaces the existing thesaurus with the thesaurus specified in the updated path.

It can take up to 30 minutes to see the effects of your updated thesaurus file.

> **Note**
>
> If there are validation or syntax errors in the thesaurus file, the previously uploaded thesaurus file is retained.

The following procedures show how to modify thesaurus details.

Console

To modify thesaurus details

1. In the left navigation pane, under the index you want to modify, choose **Synonyms**.
2. On the **Synonym** page, select the thesaurus you want to modify and then choose **Edit**.
3. On the **Update thesaurus** page, update the thesaurus details.
4. (Optional) Choose **Change the thesaurus file path** and then specify an Amazon S3 path to the new thesaurus file. Your existing thesaurus file is replaced by the file you specify. If you do not change the path, Amazon Kendra reloads the thesaurus from the existing path.

   If you select **Keep the current thesaurus file**, Amazon Kendra does not reload the thesaurus file.
5. Choose **Save** to save the configuration.

You can also reload the thesaurus from the existing thesaurus path.

To reload a thesaurus from an existing path

1. In the left navigation pane, under the index you want to modify, choose **Synonyms**.
2. On the **Synonym** page, select the thesaurus you want to reload and then choose **Refresh**.
3. On the **Reload thesaurus file** page, confirm you want to refresh the thesaurus file.

CLI

To update a thesaurus, call `update-thesaurus`:

```
aws kendra update-thesaurus \
--index-id index-id \
--name "thesaurus-name" \
```

```
--description "thesaurus-description" \
--source-s3-path "Bucket=bucket-name,Key=thesaurus/synonyms.txt" \
--role-arn role-arn
```

Python

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Update a thesaurus")

thesaurus_name = "thesaurus-name"
thesaurus_description = "thesaurus-description"
thesaurus_role_arn = "role-arn"

thesaurus_id = "thesaurus-id"
index_id = "index-id"

s3_bucket_name = "bucket-name"
s3_key = "thesaurus-file"
source_s3_path= {
    'Bucket': s3_bucket_name,
    'Key': s3_key
}

try:
    kendra.update_thesaurus(
        Id = thesaurus_id,
        IndexId = index_id,
        Description = thesaurus_description,
        Name = thesaurus_name,
        RoleArn = thesaurus_role_arn,
        SourceS3Path = source_s3_path
    )

    print("Wait for Kendra to update the thesaurus.")

    while True:
        # Get thesaurus description
```

```
        thesaurus_description = kendra.describe_thesaurus(
            Id = thesaurus_id,
            IndexId = index_id
        )
        # If status is not UPDATING quit
        status = thesaurus_description["Status"]
        print("Updating thesaurus. Status: " + status)
        if status != "UPDATING":
            break
        time.sleep(60)

except ClientError as e:
        print("%s" % e)

print("Program ends.")
```

Java

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.UpdateThesaurusRequest;
import software.amazon.awssdk.services.kendra.model.DescribeThesaurusRequest;
import software.amazon.awssdk.services.kendra.model.DescribeThesaurusResponse;
import software.amazon.awssdk.services.kendra.model.S3Path;
import software.amazon.awssdk.services.kendra.model.ThesaurusStatus;

public class UpdateThesaurusExample {

  public static void main(String[] args) throws InterruptedException {

    KendraClient kendra = KendraClient.builder().build();

    String thesaurusName = "thesaurus-name";
    String thesaurusDescription = "thesaurus-description";
    String thesaurusRoleArn = "role-arn";

    String s3BucketName = "bucket-name";
    String s3Key = "thesaurus-file";

    String thesaurusId = "thesaurus-id";
    String indexId = "index-id";
```

```
    UpdateThesaurusRequest updateThesaurusRequest = UpdateThesaurusRequest
        .builder()
        .id(thesaurusId)
        .indexId(indexId)
        .name(thesaurusName)
        .description(thesaurusDescription)
        .roleArn(thesaurusRoleArn)
        .sourceS3Path(S3Path.builder()
            .bucket(s3BucketName)
            .key(s3Key)
            .build())
        .build();
    kendra.updateThesaurus(updateThesaurusRequest);

    System.out.println(String.format("Waiting until the thesaurus with ID %s is
 updated.", thesaurusId));

    // a new source s3 path requires re-consumption by Kendra
    // and so can take as long as a Create Thesaurus operation
    while (true) {
      DescribeThesaurusRequest describeThesaurusRequest =
 DescribeThesaurusRequest.builder()
          .id(thesaurusId)
          .indexId(indexId)
          .build();
      DescribeThesaurusResponse describeThesaurusResponse =
 kendra.describeThesaurus(describeThesaurusRequest);
      ThesaurusStatus status = describeThesaurusResponse.status();
      if (status != ThesaurusStatus.UPDATING) {
        break;
      }

      TimeUnit.SECONDS.sleep(60);
    }

    System.out.println("Thesaurus update is complete.");
  }
}
```

Deleting a thesaurus

The following procedures show how to delete a thesaurus.

Console

1. In the left navigation pane, under the index you want to modify, choose **Synonyms**.
2. On the **Synonym** page, select the thesaurus you want to delete.
3. On the **Thesaurus detail** page, choose **Delete** and then confirm to delete.

CLI

To delete a thesarus to an index with the AWS CLI, call `delete-thesaurus`:

```
aws kendra delete-thesaurus \
--index-id index-id \
--id thesaurus-id
```

Python

```
import boto3
from botocore.exceptions import ClientError

kendra = boto3.client("kendra")

print("Delete a thesaurus")

thesaurus_id = "thesaurus-id"
index_id = "index-id"

try:
    kendra.delete_thesaurus(
        Id = thesaurus_id,
        IndexId = index_id
    )

except ClientError as e:
        print("%s" % e)

print("Program ends.")
```

Java

```
package com.amazonaws.kendra;
```

```
import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.DeleteThesaurusRequest;

public class DeleteThesaurusExample {

  public static void main(String[] args) throws InterruptedException {

    KendraClient kendra = KendraClient.builder().build();

    String thesaurusId = "thesaurus-id";
    String indexId = "index-id";

    DeleteThesaurusRequest updateThesaurusRequest = DeleteThesaurusRequest
        .builder()
        .id(thesaurusId)
        .indexId(indexId)
        .build();
    kendra.deleteThesaurus(updateThesaurusRequest);
  }
}
```

Highlights in search results

Synonym highlighting is on by default. Highlight information is included in Amazon Kendra SDK and CLI query results. If you interact with Amazon Kendra using the SDK or CLI, you determine how to display results.

Synonym highlights will have the highlight type `THESAURUS_SYNONYM`. For more information about highlights, see the Highlight object.

Tutorial: Building a metadata-enriched, intelligent search solution with Amazon Kendra

This tutorial shows you how to build a metadata-enriched, natural language based, intelligent search solution for your enterprise data using Amazon Kendra, Amazon Comprehend, Amazon Simple Storage Service (S3), and AWS CloudShell.

Amazon Kendra is an intelligent search service that can build a search index for your unstructured, natural language data repositories. To make it easier for your customers to find and filter relevant answers, you can use Amazon Comprehend to extract metadata from your data and ingest it into your Amazon Kendra search index.

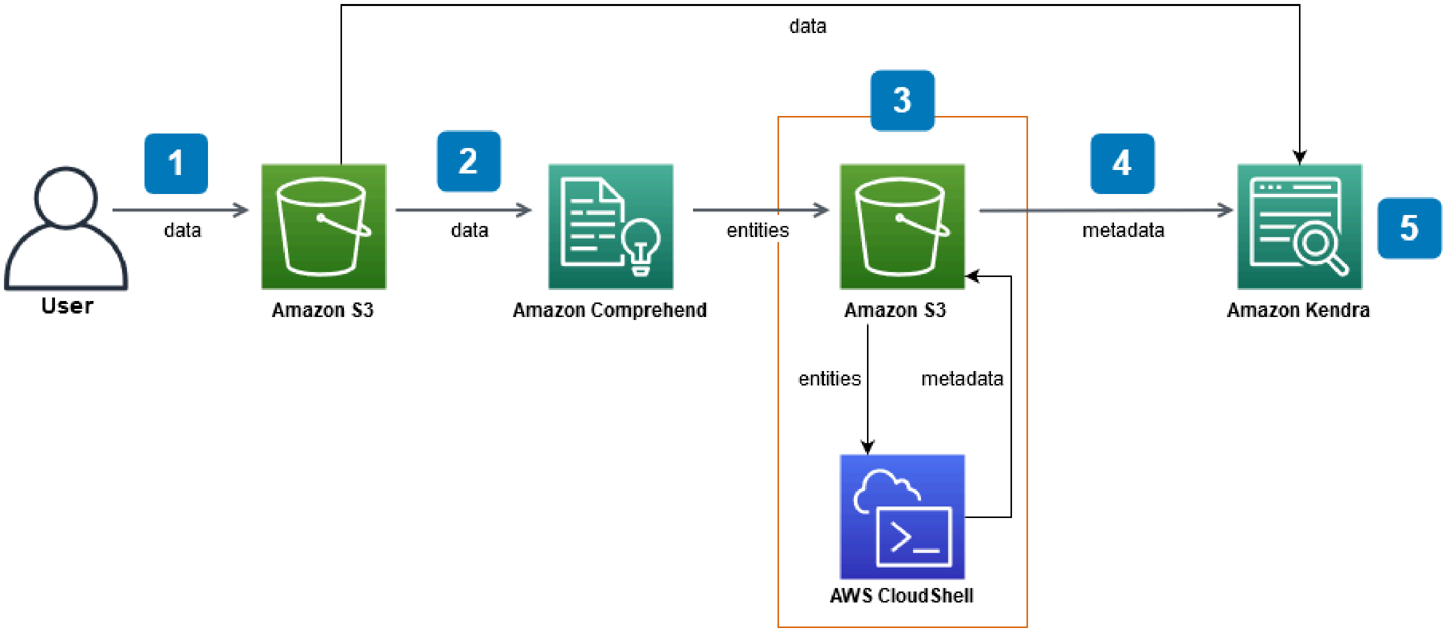
Amazon Comprehend is a natural language processing (NLP) service that can identify entities. Entities are references to people, places, locations, organizations, and objects in your data.

This tutorial uses a sample dataset of news articles to extract entities, convert them to metadata, and ingest them into your Amazon Kendra index to run searches on. The added metadata lets you filter your search results using any subset of these entities, and improves search accuracy. By following this tutorial, you will learn how to create a search solution for your enterprise data without any specialized machine learning knowledge.

This tutorial shows you how to build your search solution using the following steps:

1. Storing a sample dataset of news articles in Amazon S3.
2. Using Amazon Comprehend to extract entities from your data.
3. Running a Python 3 script to convert the entities into Amazon Kendra index metadata format and storing this metadata in S3.
4. Creating an Amazon Kendra search index and ingesting the data and the metadata.
5. Querying the search index.

The following diagram shows the workflow:



Estimated time to complete this tutorial: 1 hour

Estimated cost: Some of the actions in this tutorial incur charges on your AWS account. For more information on the cost of each service, see the price pages for Amazon S3, Amazon Comprehend, AWS CloudShell, and Amazon Kendra.

Topics

Prerequisites

To complete this tutorial, you need the following resources:

- An AWS account. If you do not have an AWS account, follow the steps in Setting up Amazon Kendra to set up your AWS account.

- A development computer running Windows, macOS, or Linux, to access the AWS Management Console. For more information, see Configuring the AWS Management Console.
- An AWS Identity and Access Management (IAM) user. To learn how to set up an IAM user and group for your account, see the Getting Started section in the *IAM User Guide*.

  If you are using the AWS Command Line Interface, you also need to attach the following policy to your IAM user to grant it the basic permissions required to complete this tutorial.

  For more information, see Creating IAM policies and Adding and removing IAM identity permissions.
- The AWS Regional Services List. To reduce latency, you should choose the AWS region closest to your geographic location that is supported by both Amazon Comprehend and Amazon Kendra.
- (Optional) An AWS Key Management Service. While this tutorial does not use encryption, you might want to use encryption best practices for your specific use case.
- (Optional) An Amazon Virtual Private Cloud. While this tutorial does not use a VPC, you might want to use VPC best practices to ensure data security for your specific use case.

Step 1: Adding documents to Amazon S3

Before you run an Amazon Comprehend entities analysis job on your dataset, you create an Amazon S3 bucket to host the data, metadata, and the Amazon Comprehend entities analysis output.

Topics

- Downloading the sample dataset
- Creating an Amazon S3 bucket
- Creating data and metadata folders in your S3 bucket
- Uploading the input data

Downloading the sample dataset

Before Amazon Comprehend can run an entities analysis job on your data, you must download and extract the dataset and upload it to an S3 bucket.

To download and extract the dataset (Console)

1. Download the tutorial-dataset.zip folder on your device.
2. Extract the `tutorial-dataset` folder to access the `data` folder.

To download and extract the dataset (Terminal)

1. To download the `tutorial-dataset`, run the following command on a terminal window:

   Linux

   ```
   curl -o path/tutorial-dataset.zip https://docs.aws.amazon.com/kendra/latest/dg/
   samples/tutorial-dataset.zip
   ```

   Where:

   - *path/* is the local filepath to the location you want to save the zip folder in.

   macOS

   ```
   curl -o path/tutorial-dataset.zip https://docs.aws.amazon.com/kendra/latest/dg/
   samples/tutorial-dataset.zip
   ```

   Where:

   - *path/* is the local filepath to the location you want to save the zip folder in.

   Windows

   ```
   curl -o path/tutorial-dataset.zip https://docs.aws.amazon.com/kendra/latest/dg/
   samples/tutorial-dataset.zip
   ```

   Where:

   - *path/* is the local filepath to the location you want to save the zip folder in.
2. To extract the data from the zip folder, run the following command on the terminal window:

Linux

```
unzip path/tutorial-dataset.zip -d path/
```

Where:

- *path/* is the local filepath to your saved zip folder.

macOS

```
unzip path/tutorial-dataset.zip -d path/
```

Where:

- *path/* is the local filepath to your saved zip folder.

Windows

```
tar -xf path/tutorial-dataset.zip -C path/
```

Where:

- *path/* is the local filepath to your saved zip folder.

At the end of this step, you should have the extracted files in a decompressed folder called `tutorial-dataset`. This folder contains a `README` file with an Apache 2.0 open source attribution and a folder called `data` containing the dataset for this tutorial. The dataset consists of 100 files with `.story` extensions.

Creating an Amazon S3 bucket

After downloading and extracting the sample data folder, you store it in an Amazon S3 bucket.

Important

The name of an Amazon S3 bucket must be unique across all of AWS.

To create an S3 bucket (Console)

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In **Buckets**, choose **Create bucket**.
3. For **Bucket name**, enter a unique name.
4. For **Region**, choose the AWS region where you want to create the bucket.

> **Note**
>
> You must choose a region that supports both Amazon Comprehend and Amazon Kendra. You cannot change the region of a bucket after you have created it.

5. Keep the default settings for **Block Public Access settings for this bucket**, **Bucket Versioning**, and **Tags**.
6. For **Default encryption**, choose **Disable**.
7. Keep the default settings for the **Advanced settings**.
8. Review your bucket configuration and then choose **Create bucket**.

To create an S3 bucket (AWS CLI)

1. To create an S3 bucket, use the create-bucket command in the AWS CLI:

   Linux

   ```
   aws s3api create-bucket \
           --bucket amzn-s3-demo-bucket \
           --region aws-region \
           --create-bucket-configuration LocationConstraint=aws-region
   ```

   Where:

   - amzn-s3-demo-bucket is your bucket name,
   - *aws-region* is the region you want to create your bucket in.

macOS

```
aws s3api create-bucket \
        --bucket amzn-s3-demo-bucket \
        --region aws-region \
        --create-bucket-configuration LocationConstraint=aws-region
```

Where:

- amzn-s3-demo-bucket is your bucket name,
- *aws-region* is the region you want to create your bucket in.

Windows

```
aws s3api create-bucket ^
        --bucket amzn-s3-demo-bucket ^
        --region aws-region ^
        --create-bucket-configuration LocationConstraint=aws-region
```

Where:

- amzn-s3-demo-bucket is your bucket name,
- *aws-region* is the region you want to create your bucket in.

> **Note**
>
> You must choose a region that supports both Amazon Comprehend and Amazon Kendra. You cannot change the region of a bucket after you have created it.

2. To ensure that your bucket was created successfully, use the list command:

Linux

```
aws s3 ls
```

macOS

```
aws s3 ls
```

Windows

```
aws s3 ls
```

Creating data and metadata folders in your S3 bucket

After creating your S3 bucket, you create data and metadata folders inside it.

To create folders in your S3 bucket (Console)

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In **Buckets**, click on the name of your bucket from the list of buckets.
3. From the **Objects** tab, choose **Create folder**.
4. For the new folder name, enter **`data`**.
5. For the encryption settings, choose **Disable**.
6. Choose **Create folder**.
7. Repeat steps 3 to 6 to create another folder for storing the Amazon Kendra metadata and name the folder created in step 4 **`metadata`**.

To create folders in your S3 bucket (AWS CLI)

1. To create the `data` folder in your S3 bucket, use the put-object command in the AWS CLI:

   Linux

   ```
   aws s3api put-object \
           --bucket amzn-s3-demo-bucket \
           --key data/
   ```

   Where:

   - amzn-s3-demo-bucket is your bucket name.

macOS

```
aws s3api put-object \
        --bucket amzn-s3-demo-bucket \
        --key data/
```

Where:

- amzn-s3-demo-bucket is your bucket name.

Windows

```
aws s3api put-object ^
        --bucket amzn-s3-demo-bucket ^
        --key data/
```

Where:

- amzn-s3-demo-bucket is your bucket name.

2. To create the `metadata` folder in your S3 bucket, use the put-object command in the AWS CLI:

Linux

```
aws s3api put-object \
        --bucket amzn-s3-demo-bucket \
        --key metadata/
```

Where:

- amzn-s3-demo-bucket is your bucket name.

macOS

```
aws s3api put-object \
        --bucket amzn-s3-demo-bucket \
        --key metadata/
```

Where:

- amzn-s3-demo-bucket is your bucket name.

Windows

```
aws s3api put-object ^
        --bucket amzn-s3-demo-bucket ^
        --key metadata/
```

Where:

- amzn-s3-demo-bucket is your bucket name.

3. To ensure that your folders were created successfully, check the contents of your bucket using the list command:

   Linux

```
aws s3 ls s3://amzn-s3-demo-bucket/
```

Where:

- amzn-s3-demo-bucket is your bucket name.

macOS

```
aws s3 ls s3://amzn-s3-demo-bucket/
```

Where:

- amzn-s3-demo-bucket is your bucket name.

Windows

```
aws s3 ls s3://amzn-s3-demo-bucket/
```

Where:

- amzn-s3-demo-bucket is your bucket name.

Uploading the input data

After creating your data and metadata folders, you upload the sample dataset into the `data` folder.

To upload the sample dataset into the data folder (Console)

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In **Buckets**, click on the name of your bucket from the list of buckets and then click on `data`.
3. Choose **Upload** and then choose **Add files**.
4. In the dialog box, navigate to the `data` folder inside the `tutorial-dataset` folder in your local device, select all the files, and then choose **Open**.
5. Keep the default settings for **Destination**, **Permissions**, and **Properties**.
6. Choose **Upload**.

To upload the sample dataset into the data folder (AWS CLI)

1. To upload the sample data into the `data` folder, use the copy command in the AWS CLI:

   Linux

   ```
   aws s3 cp path/tutorial-dataset/data s3://amzn-s3-demo-bucket/data/ --recursive
   ```

   Where:

   - *path/* is the filepath to the `tutorial-dataset` folder on your device,
   - amzn-s3-demo-bucket is your bucket name.

   macOS

   ```
   aws s3 cp path/tutorial-dataset/data s3://amzn-s3-demo-bucket/data/ --recursive
   ```

   Where:

- *path/* is the filepath to the `tutorial-dataset` folder on your device,
- amzn-s3-demo-bucket is your bucket name.

Windows

```
aws s3 cp path/tutorial-dataset/data s3://amzn-s3-demo-bucket/data/ --recursive
```

Where:

- *path/* is the filepath to the `tutorial-dataset` folder on your device,
- amzn-s3-demo-bucket is your bucket name.

2. To ensure that your dataset files were uploaded successfully to your `data` folder, use the [list](https://docs.aws.amazon.com/cli/latest/reference/s3/ls.html) command in the AWS CLI:

Linux

```
aws s3 ls s3://amzn-s3-demo-bucket/data/
```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket.

macOS

```
aws s3 ls s3://amzn-s3-demo-bucket/data/
```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket.

Windows

```
aws s3 ls s3://amzn-s3-demo-bucket/data/
```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket.

At the end of this step, you have an S3 bucket with your dataset stored inside the `data` folder, and an empty `metadata` folder, which will store your Amazon Kendra metadata.

Step 2: Running an entities analysis job on Amazon Comprehend

After storing the sample dataset in your S3 bucket, you run an Amazon Comprehend entities analysis job to extract entities from your documents. These entities will form Amazon Kendra custom attributes and help you filter search results on your index. For more information, see Detect Entities.

Topics

Running an Amazon Comprehend entities analysis job

To extract entities from your dataset, you run an Amazon Comprehend entities analysis job.

If you are using the AWS CLI in this step, you first create and attach an AWS IAM role and policy for Amazon Comprehend and then run an entities analysis job. To run an entities analysis job on your sample data, Amazon Comprehend needs:

- an AWS Identity and Access Management (IAM) role that recognizes it as a trusted entity
- an AWS IAM policy attached to the IAM role that gives it permissions to access your S3 bucket

For more information, see How Amazon Comprehend works with IAM and Identity-Based Policies for Amazon Comprehend.

To run an Amazon Comprehend entities analysis job (Console)

1. Open the Amazon Comprehend console at https://console.aws.amazon.com/comprehend/.

> **Important**
>
> Ensure that you are in the same region in which you created your Amazon S3 bucket. If you are in another region, choose the AWS region where you created your S3 bucket from the **Region selector** in the top navigation bar.

2. Choose **Launch Amazon Comprehend**.
3. In the left navigation pane, choose **Analysis jobs**.
4. Choose **Create job**.
5. In the **Job settings** section, do the following:
   a. For **Name**, enter **`data-entities-analysis`**.
   b. For **Analysis type**, choose **Entities**.
   c. For **Language**, choose **English**.
   d. Keep **Job encryption** turned off.
6. In the **Input data** section, do the following:
   a. For **Data source**, choose **My documents**.
   b. For **S3 location**, choose **Browse S3**.
   c. For **Choose resources**, click on the name of your bucket from the list of buckets.
   d. For **Objects**, select the option button for `data` and choose **Choose**.
   e. For **Input format**, choose **One document per file**.
7. In the **Output data** section, do the following:
   a. For **S3 location**, choose **Browse S3** and then select the option box for your bucket from the list of buckets and choose **Choose**.
   b. Keep **Encryption** turned off.
8. In the **Access permissions** section, do the following:
   a. For **IAM role**, choose **Create an IAM role**.
   b. For **Permissions to access**, choose **Input and Output S3 buckets**.
   c. For **Name suffix**, enter **`comprehend-role`**. This role provides access to your Amazon S3 bucket.
9. Keep the default **VPC settings**.

10. Choose **Create job**.

To run an Amazon Comprehend entities analysis job (AWS CLI)

1. To create and attach an IAM role for Amazon Comprehend that recognizes it as a trusted entity, do the following:

   a. Save the following trust policy as a JSON file called `comprehend-trust-policy.json` in a text editor on your local device.

      JSON

      ```
      {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Principal": {
              "Service": "comprehend.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
          }
        ]
      }
      ```

   b. To create an IAM role called `comprehend-role` and attach your saved `comprehend-trust-policy.json` file to it, use the create-role command:

      Linux

      ```
      aws iam create-role \
                --role-name comprehend-role \
                --assume-role-policy-document file://path/comprehend-trust-
      policy.json
      ```

      Where:

      - *path/* is the filepath to `comprehend-trust-policy.json` on your local device.

macOS

```
aws iam create-role \
          --role-name comprehend-role \
          --assume-role-policy-document file://path/comprehend-trust-policy.json
```

Where:

- *path/* is the filepath to `comprehend-trust-policy.json` on your local device.

Windows

```
aws iam create-role ^
          --role-name comprehend-role ^
          --assume-role-policy-document file://path/comprehend-trust-policy.json
```

Where:

- *path/* is the filepath to `comprehend-trust-policy.json` on your local device.

c. Copy the Amazon Resource Name (ARN) to your text editor and save it locally as `comprehend-role-arn`.

> **Note**
>
> The ARN has a format similar to *arn:aws:iam::123456789012:role/comprehend-role*. You need the ARN you saved as `comprehend-role-arn` to run the Amazon Comprehend analysis job.

2. To create and attach an IAM policy to your IAM role that grants it permissions to access your S3 bucket, do the following:

   a. Save the following trust policy as a JSON file called `comprehend-S3-access-policy.json` in a text editor on your local device.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

b. To create an IAM policy called `comprehend-S3-access-policy` to access your S3 bucket, use the [create-policy](https://docs.aws.amazon.com/cli/latest/reference/iam/create-policy.html) command:

Linux

```
aws iam create-policy \
```

```
            --policy-name comprehend-S3-access-policy \
            --policy-document file://path/comprehend-S3-access-policy.json
```

Where:

- *path/* is the filepath to `comprehend-S3-access-policy.json` on your local device.

macOS

```
aws iam create-policy \
            --policy-name comprehend-S3-access-policy \
            --policy-document file://path/comprehend-S3-access-policy.json
```

Where:

- *path/* is the filepath to `comprehend-S3-access-policy.json` on your local device.

Windows

```
aws iam create-policy ^
            --policy-name comprehend-S3-access-policy ^
            --policy-document file://path/comprehend-S3-access-policy.json
```

Where:

- *path/* is the filepath to `comprehend-S3-access-policy.json` on your local device.

c. Copy the Amazon Resource Name (ARN) to your text editor and save it locally as `comprehend-S3-access-arn`.

> **Note**
>
> The ARN has a format similar to *`arn:aws:iam::123456789012:role/comprehend-S3-access-policy`*. You need the ARN you saved as

> `comprehend-S3-access-arn` to attach the `comprehend-S3-access-policy` to your IAM role.

d. To attach the `comprehend-S3-access-policy` to your IAM role, use the [attach-role-policy](#) command:

Linux

```
aws iam attach-role-policy \
          --policy-arn policy-arn \
          --role-name comprehend-role
```

Where:

- *`policy-arn`* is the ARN you saved as `comprehend-S3-access-arn`.

macOS

```
aws iam attach-role-policy \
          --policy-arn policy-arn \
          --role-name comprehend-role
```

Where:

- *`policy-arn`* is the ARN you saved as `comprehend-S3-access-arn`.

Windows

```
aws iam attach-role-policy ^
          --policy-arn policy-arn ^
          --role-name comprehend-role
```

Where:

- *`policy-arn`* is the ARN you saved as `comprehend-S3-access-arn`.

3. To run an Amazon Comprehend entities analysis job, use the [start-entities-detection-job](#) command:

Linux

```
aws comprehend start-entities-detection-job \
        --input-data-config S3Uri=s3://amzn-s3-demo-bucket/
data/,InputFormat=ONE_DOC_PER_FILE \
        --output-data-config S3Uri=s3://amzn-s3-demo-bucket/ \
        --data-access-role-arn role-arn \
        --job-name data-entities-analysis \
        --language-code en \
        --region aws-region
```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket,
- *role-arn* is the ARN you saved as `comprehend-role-arn`,
- *aws-region* is your AWS region.

macOS

```
aws comprehend start-entities-detection-job \
        --input-data-config S3Uri=s3://amzn-s3-demo-bucket/
data/,InputFormat=ONE_DOC_PER_FILE \
        --output-data-config S3Uri=s3://amzn-s3-demo-bucket/ \
        --data-access-role-arn role-arn \
        --job-name data-entities-analysis \
        --language-code en \
        --region aws-region
```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket,
- *role-arn* is the ARN you saved as `comprehend-role-arn`,
- *aws-region* is your AWS region.

Windows

```
aws comprehend start-entities-detection-job ^
```

```
        --input-data-config S3Uri=s3://amzn-s3-demo-bucket/
data/,InputFormat=ONE_DOC_PER_FILE ^
        --output-data-config S3Uri=s3://amzn-s3-demo-bucket/ ^
        --data-access-role-arn role-arn ^
        --job-name data-entities-analysis ^
        --language-code en ^
        --region aws-region
```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket,
- *role-arn* is the ARN you saved as `comprehend-role-arn`,
- *aws-region* is your AWS region.

4. Copy the entities analysis `JobId` and save it in a text editor as `comprehend-job-id`. The `JobId` helps you track the status of your entities analysis job.
5. To track the progress of your entities analysis job, use the describe-entities-detection-job command:

   Linux

   ```
   aws comprehend describe-entities-detection-job \
           --job-id entities-job-id \
           --region aws-region
   ```

   Where:

   - *entities-job-id* is your saved `comprehend-job-id`,
   - *aws-region* is your AWS region.

   macOS

   ```
   aws comprehend describe-entities-detection-job \
           --job-id entities-job-id \
           --region aws-region
   ```

   Where:

   - *entities-job-id* is your saved `comprehend-job-id`,

- *aws-region* is your AWS region.

Windows

```
aws comprehend describe-entities-detection-job ^
        --job-id entities-job-id ^
        --region aws-region
```

Where:

- *entities-job-id* is your saved `comprehend-job-id`,
- *aws-region* is your AWS region.

It can take several minutes for the `JobStatus` to change to `COMPLETED`.

At the end of this step, Amazon Comprehend stores the entity analysis results as a zipped `output.tar.gz` file inside an `output` folder within an auto-generated folder in your S3 bucket. Make sure that your analysis job status is complete before you move on to the next step.

Step 3: Formatting the entities analysis output as Amazon Kendra metadata

To convert the entities extracted by Amazon Comprehend to the metadata format required by an Amazon Kendra index, you run a Python 3 script. The results of the conversion are stored in the `metadata` folder in your Amazon S3 bucket.

For more information on Amazon Kendra metadata format and structure, see S3 document metadata.

Topics

Downloading and extracting the Amazon Comprehend output

To format the Amazon Comprehend entities analysis output, you must first download the Amazon Comprehend entities analysis `output.tar.gz` archive and extract the entities analysis file.

To download and extract the output file (Console)

1. In the Amazon Comprehend console navigation pane, navigate to **Analysis jobs**.
2. Choose your entities analysis job `data-entities-analysis`.
3. Under **Output**, choose the link displayed next to **Output data location**. This redirects you to the `output.tar.gz` archive in your S3 bucket.
4. In the **Overview** tab, choose **Download**.

> **Tip**
>
> The output of all Amazon Comprehend analysis jobs have the same name. Renaming your archive will help you track it more easily.

5. Decompress and extract the downloaded Amazon Comprehend file to your device.

To download and extract the output file (AWS CLI)

1. To access the name of the Amazon Comprehend auto-generated folder in your S3 bucket which contains the results of the entities analysis job, use the describe-entities-detection-job command:

   Linux

   ```
   aws comprehend describe-entities-detection-job \
             --job-id entities-job-id \
             --region aws-region
   ```

   Where:

   - *`entities-job-id`* is your saved `comprehend-job-id` from the section called "Step 2: Detecting entities",
   - *`aws-region`* is your AWS region.

macOS

```
aws comprehend describe-entities-detection-job \
          --job-id entities-job-id \
          --region aws-region
```

Where:

- *entities-job-id* is your saved `comprehend-job-id` from [the section called "Step 2: Detecting entities"](),
- *aws-region* is your AWS region.

Windows

```
aws comprehend describe-entities-detection-job ^
          --job-id entities-job-id ^
          --region aws-region
```

Where:

- *entities-job-id* is your saved `comprehend-job-id` from [the section called "Step 2: Detecting entities"](),
- *aws-region* is your AWS region.

2. From the `OutputDataConfig` object in your entities job description, copy and save the `S3Uri` value as `comprehend-S3uri` on a text editor.

> **Note**
>
> The `S3Uri` value has a format similar to *s3://amzn-s3-demo-bucket/.../output/output.tar.gz*.

3. To download the entities output archive, use the [copy]() command:

Linux

```
aws s3 cp s3://amzn-s3-demo-bucket/.../output/output.tar.gz path/output.tar.gz
```

Where:

- *s3://amzn-s3-demo-bucket/.../output/output.tar.gz* is the `S3Uri` value you saved as `comprehend-S3uri`,
- *path/* is the local directory where you wish to save the output.

macOS

```
aws s3 cp s3://amzn-s3-demo-bucket/.../output/output.tar.gz path/output.tar.gz
```

Where:

- *s3://amzn-s3-demo-bucket/.../output/output.tar.gz* is the `S3Uri` value you saved as `comprehend-S3uri`,
- *path/* is the local directory where you wish to save the output.

Windows

```
aws s3 cp s3://amzn-s3-demo-bucket/.../output/output.tar.gz path/output.tar.gz
```

Where:

- *s3://amzn-s3-demo-bucket/.../output/output.tar.gz* is the `S3Uri` value you saved as `comprehend-S3uri`,
- *path/* is the local directory where you wish to save the output.

4. To extract the entities output, run the following command on a terminal window:

Linux

```
tar -xf path/output.tar.gz -C path/
```

Where:

- *path/* is the filepath to the downloaded `output.tar.gz` archive on your local device.

macOS

```
tar -xf path/output.tar.gz -C path/
```

Where:

- *path/* is the filepath to the downloaded `output.tar.gz` archive on your local device.

Windows

```
tar -xf path/output.tar.gz -C path/
```

Where:

- *path/* is the filepath to the downloaded `output.tar.gz` archive on your local device.

At the end of this step, you should have a file on your device called `output` with a list of Amazon Comprehend identified entities.

Uploading the output into the S3 bucket

After downloading and extracting the Amazon Comprehend entities analysis file, you upload the extracted `output` file to your Amazon S3 bucket.

To upload the extracted Amazon Comprehend output file (Console)

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In **Buckets**, click on the name of your bucket and then choose **Upload**.
3. In **Files and folders**, choose **Add files**.
4. In the dialog box, navigate to your extracted `output` file in your device, select it, and choose **Open**.
5. Keep the default settings for **Destination**, **Permissions**, and **Properties**.
6. Choose **Upload**.

To upload the extracted Amazon Comprehend output file (AWS CLI)

1. To upload the extracted `output` file to your bucket, use the [copy](https://docs.aws.amazon.com/cli/latest/reference/s3/cp.html) command:

   Linux

   ```
   aws s3 cp path/output s3://amzn-s3-demo-bucket/output
   ```

   Where:

   - *path/* is the local filepath to your extracted `output` file,
   - amzn-s3-demo-bucket is the name of your S3 bucket.

   macOS

   ```
   aws s3 cp path/output s3://amzn-s3-demo-bucket/output
   ```

   Where:

   - *path/* is the local filepath to your extracted `output` file,
   - amzn-s3-demo-bucket is the name of your S3 bucket.

   Windows

   ```
   aws s3 cp path/output s3://amzn-s3-demo-bucket/output
   ```

   Where:

   - *path/* is the local filepath to your extracted `output` file,
   - amzn-s3-demo-bucket is the name of your S3 bucket.

2. To ensure that the `output` file was uploaded successfully to your S3 bucket, check its contents by using the [list](https://docs.aws.amazon.com/cli/latest/reference/s3/ls.html) command:

   Linux

   ```
   aws s3 ls s3://amzn-s3-demo-bucket/
   ```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket.

macOS

```
aws s3 ls s3://amzn-s3-demo-bucket/
```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket.

Windows

```
aws s3 ls s3://amzn-s3-demo-bucket/
```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket.

Converting the output to Amazon Kendra metadata format

To convert the Amazon Comprehend output to Amazon Kendra metadata, you run a Python 3 script. If you are using the Console, you use AWS CloudShell for this step.

To run the Python 3 script (Console)

1. Download the converter.py.zip zipped file on your device.
2. Extract the Python 3 file `converter.py`.
3. Sign into the AWS Management Console and make sure your AWS region is set to the same region as your S3 bucket and your Amazon Comprehend analysis job.
4. Choose the **AWS CloudShell icon** or type **AWS CloudShell** in the **Search** box on the top navigation bar to launch an environment.

> **Note**
>
> When AWS CloudShell launches in a new browser window for the first time, a welcome panel displays and lists key features. The shell is ready for interaction after you close this panel and the command prompt displays.

5. After the terminal is prepared, choose **Actions** from the navigation pane and then choose **Upload file** from the menu.
6. In the dialog box that opens, choose **Select file** and then choose the downloaded Python 3 file `converter.py` from your device. Choose **Upload**.
7. In the AWS CloudShell environment, enter the following command:

```
python3 converter.py
```

8. When the shell interface prompts you to **Enter the name of your S3 bucket**, enter the name of your S3 bucket and press enter.
9. When the shell interface prompts you to **Enter the full filepath to your Comprehend output file**, enter **`output`** and press enter.
10. When the shell interface prompts you to **Enter the full filepath to your metadata folder**, enter **`metadata/`** and press enter.

> **Important**
>
> For the metadata to be formatted correctly, the input values in steps 8-10 must be exact.

To run the Python 3 script (AWS CLI)

1. To download the Python 3 file `converter.py`, run the following command on a terminal window:

   Linux

   ```
   curl -o path/converter.py.zip https://docs.aws.amazon.com/kendra/latest/dg/
   samples/converter.py.zip
   ```

Where:

- *path/* is the filepath to the location you want to save the zipped file in.

macOS

```
curl -o path/converter.py.zip https://docs.aws.amazon.com/kendra/latest/dg/
samples/converter.py.zip
```

Where:

- *path/* is the filepath to the location you want to save the zipped file in.

Windows

```
curl -o path/converter.py.zip https://docs.aws.amazon.com/kendra/latest/dg/
samples/converter.py.zip
```

Where:

- *path/* is the filepath to the location you want to save the zipped file in.

2. To extract the Python 3 file, run the following command on the terminal window:

Linux

```
unzip path/converter.py.zip -d path/
```

Where:

- *path/* is the filepath to your saved `converter.py.zip`.

macOS

```
unzip path/converter.py.zip -d path/
```

Where:

- *path/* is the filepath to your saved `converter.py.zip`.

Windows

```
tar -xf path/converter.py.zip -C path/
```

Where:

- *path/* is the filepath to your saved `converter.py.zip`.

3. Make sure that Boto3 is installed on your device by running the following command.

Linux

```
pip3 show boto3
```

macOS

```
pip3 show boto3
```

Windows

```
pip3 show boto3
```

Note

If you do not have Boto3 installed, run `pip3 install boto3` to install it.

4. To run the Python 3 script to convert the `output` file, run the following command.

Linux

```
python path/converter.py
```

Where:

- *path/* is the filepath to your saved `converter.py.zip`.

macOS

```
python path/converter.py
```

Where:

- *path/* is the filepath to your saved `converter.py.zip`.

Windows

```
python path/converter.py
```

Where:

- *path/* is the filepath to your saved `converter.py.zip`.

5. When the AWS CLI prompts you to `Enter the name of your S3 bucket`, enter the name of your S3 bucket and press enter.
6. When the AWS CLI prompts you to `Enter the full filepath to your Comprehend output file`, enter **`output`** and press enter.
7. When the AWS CLI prompts you to `Enter the full filepath to your metadata folder`, enter **`metadata/`** and press enter.

> **Important**
>
> For the metadata to be formatted correctly, the input values in steps 5-7 must be exact.

At the end of this step, the formatted metadata is deposited inside the `metadata` folder in your S3 bucket.

Cleaning up your Amazon S3 bucket

Since the Amazon Kendra index syncs all files stored in a bucket, we recommend you clean up your Amazon S3 bucket to prevent redundant search results.

To clean up your Amazon S3 bucket (Console)

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In **Buckets**, choose your bucket and then select the Amazon Comprehend entity analysis output folder, the Amazon Comprehend entity analysis `.temp` file, and the extracted Amazon Comprehend `output` file.
3. From the **Overview** tab choose **Delete**.
4. In **Delete objects**, choose **Permanently delete objects?** and enter **`permanently delete`** in the text input field.
5. Choose **Delete objects**.

To clean up your Amazon S3 bucket (AWS CLI)

1. To delete all files and folders in your S3 bucket except the `data` and `metadata` folders, use the [remove](https://docs.aws.amazon.com/cli/latest/reference/s3/rm.html) command in the AWS CLI:

   Linux

   ```
   aws s3 rm s3://amzn-s3-demo-bucket/ --recursive --exclude "data/*" --exclude
    "metadata/*"
   ```

   Where:

   - amzn-s3-demo-bucket is the name of your S3 bucket.

   macOS

   ```
   aws s3 rm s3://amzn-s3-demo-bucket/ --recursive --exclude "data/*" --exclude
    "metadata/*"
   ```

   Where:

   - amzn-s3-demo-bucket is the name of your S3 bucket.

Windows

```
aws s3 rm s3://amzn-s3-demo-bucket/ --recursive --exclude "data/*" --exclude
 "metadata/*"
```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket.

2. To ensure that the objects were successfully deleted from your S3 bucket, check its contents by using the list command:

Linux

```
aws s3 ls s3://amzn-s3-demo-bucket/
```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket.

macOS

```
aws s3 ls s3://amzn-s3-demo-bucket/
```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket.

Windows

```
aws s3 ls s3://amzn-s3-demo-bucket/
```

Where:

- amzn-s3-demo-bucket is the name of your S3 bucket.

At the end of this step, you have converted the Amazon Comprehend entities analysis output to Amazon Kendra metadata. You are now ready to create an Amazon Kendra index.

Step 4: Creating an Amazon Kendra index and ingesting the metadata

To implement your intelligent search solution, you create an Amazon Kendra index and ingest your S3 data and metadata into it.

Before you add metadata to your Amazon Kendra index, you create custom index fields corresponding to custom document attributes, which in turn correspond to the Amazon Comprehend entity types. Amazon Kendra uses the index fields and custom document attributes you create to search and filter your documents.

For more information, see Index and Creating custom document attributes.

Topics

Creating an Amazon Kendra index

To query your source documents, you create an Amazon Kendra index.

If you are using the AWS CLI in this step, you create and attach an AWS IAM role and policy that allows Amazon Kendra to access your CloudWatch logs before creating an index. For more information, see Prerequisites.

To create an Amazon Kendra index (Console)

1. Open the Amazon Kendra console at https://console.aws.amazon.com/kendra/.

> **Important**
>
> Ensure that you are in the same region in which you created your Amazon Comprehend entities analysis job and your Amazon S3 bucket. If you are in another region, choose the AWS region where you created your Amazon S3 bucket from the **Region selector** in the top navigation bar.

2. Choose **Create an index**.
3. For **Index details** on the **Specify index details** page, do the following:

   a. For **Index name**, enter **`kendra-index`**.

   b. Keep the **Description** field blank.

   c. For **IAM role**, choose **Create a new role**. This role provides access to your Amazon S3 bucket.

   d. For **Role name**, enter **`kendra-role`**. The IAM role will have the prefix `AmazonKendra-`.

   e. Keep default settings for **Encryption** and **Tags** and choose **Next**.
4. For **Access control settings** on the **Configure user access control** page, choose **No** and then choose **Next**.
5. For **Provisioning editions** on the **Provisioning details** page, choose **Developer edition** and choose **Create**.

To create an Amazon Kendra index (AWS CLI)

1. To create and attach an IAM role for Amazon Kendra that recognizes it as a trusted entity, do the following:

   a. Save the following trust policy as a JSON file called `kendra-trust-policy.json` in a text editor on your local device.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": {
```

```
                "Service": "kendra.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
}
```

b. To create an IAM role called `kendra-role` and attach your saved `kendra-trust-policy.json` file to it, use the create-role command:

Linux

```
aws iam create-role \
          --role-name kendra-role \
          --assume-role-policy-document file://path/kendra-trust-policy.json
```

Where:

- *path/* is the filepath to `kendra-trust-policy.json` on your local device.

macOS

```
aws iam create-role \
          --role-name kendra-role \
          --assume-role-policy-document file://path/kendra-trust-policy.json
```

Where:

- *path/* is the filepath to `kendra-trust-policy.json` on your local device.

Windows

```
aws iam create-role ^
          --role-name kendra-role ^
          --assume-role-policy-document file://path/kendra-trust-policy.json
```

Where:

- *path/* is the filepath to `kendra-trust-policy.json` on your local device.

c. Copy the Amazon Resource Name (ARN) to your text editor and save it locally as `kendra-role-arn`.

> **Note**
>
> The ARN has a format similar to *`arn:aws:iam::123456789012:role/kendra-role`*. You need the ARN you saved as `kendra-role-arn` to run Amazon Kendra jobs.

2. Before you create an index, you must provide your `kendra-role` the permission to write to CloudWatch Logs. To do this, complete the following steps:

   a. Save the following trust policy as a JSON file called `kendra-cloudwatch-policy.json` in a text editor on your local device.

      Replace *`aws-region`* with your AWS region, and *`aws-account-id`* with your 12-digit AWS account ID.

   b. To create an IAM policy to access CloudWatch Logs, use the [create-policy](https://docs.aws.amazon.com/cli/latest/reference/iam/create-policy.html) command:

      Linux

```
aws iam create-policy \
          --policy-name kendra-cloudwatch-policy \
          --policy-document file://path/kendra-cloudwatch-policy.json
```

      Where:

      - *`path/`* is the filepath to `kendra-cloudwatch-policy.json` on your local device.

      macOS

```
aws iam create-policy \
          --policy-name kendra-cloudwatch-policy \
          --policy-document file://path/kendra-cloudwatch-policy.json
```

      Where:

      - *`path/`* is the filepath to `kendra-cloudwatch-policy.json` on your local device.

Windows

```
aws iam create-policy ^
          --policy-name kendra-cloudwatch-policy ^
          --policy-document file://path/kendra-cloudwatch-policy.json
```

Where:

- *path/* is the filepath to `kendra-cloudwatch-policy.json` on your local device.

c. Copy the Amazon Resource Name (ARN) to your text editor and save it locally as `kendra-cloudwatch-arn`.

> **Note**
>
> The ARN has a format similar to *arn:aws:iam::123456789012:role/kendra-cloudwatch-policy*. You need the ARN you saved as `kendra-cloudwatch-arn` to attach the `kendra-cloudwatch-policy` to your IAM role.

d. To attach the `kendra-cloudwatch-policy` to your IAM role, use the [attach-role-policy](https://docs.aws.amazon.com/cli/latest/reference/iam/attach-role-policy.html) command:

Linux

```
aws iam attach-role-policy \
          --policy-arn policy-arn \
          --role-name kendra-role
```

Where:

- *policy-arn* is your saved `kendra-cloudwatch-arn`.

macOS

```
aws iam attach-role-policy \
          --policy-arn policy-arn \
          --role-name kendra-role
```

Where:

- *policy-arn* is your saved `kendra-cloudwatch-arn`.

Windows

```
aws iam attach-role-policy ^
          --policy-arn policy-arn ^
          --role-name kendra-role
```

Where:

- *policy-arn* is your saved `kendra-cloudwatch-arn`.

3. To create an index, use the create-index command:

Linux

```
aws kendra create-index \
        --name kendra-index \
        --edition DEVELOPER_EDITION \
        --role-arn role-arn \
        --region aws-region
```

Where:

- *role-arn* is your saved `kendra-role-arn`,
- *aws-region* is your AWS region.

macOS

```
aws kendra create-index \
        --name kendra-index \
        --edition DEVELOPER_EDITION \
        --role-arn role-arn \
        --region aws-region
```

Where:

- *role-arn* is your saved kendra-role-arn,
- *aws-region* is your AWS region.

Windows

```
aws kendra create-index ^
        --name kendra-index ^
        --edition DEVELOPER_EDITION ^
        --role-arn role-arn ^
        --region aws-region
```

Where:

- *role-arn* is your saved kendra-role-arn,
- *aws-region* is your AWS region.

4. Copy the index Id and save it in a text editor as kendra-index-id. The Id helps you track the status of your index creation.
5. To track the progress of your index creation job, use the describe-index command:

Linux

```
aws kendra describe-index \
        --id kendra-index-id \
        --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

macOS

```
aws kendra describe-index \
        --id kendra-index-id \
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

Windows

```
aws kendra describe-index ^
        --id kendra-index-id ^
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

The index creation process on average takes 15 minutes, but can take longer. When the status of the index is active, your index is ready to use. While your index is being created, you can start the next step.

If you are using the AWS CLI in this step, you create and attach an IAM policy to your Amazon Kendra IAM role that gives your index permissions to access your S3 bucket.

Updating the IAM role for Amazon S3 access

While the index is being created, you update your Amazon Kendra IAM role to allow the index you created to read data from your Amazon S3 bucket. For more information, see IAM access roles for Amazon Kendra.

To update your IAM role (Console)

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the left navigation pane, choose **Roles** and enter **`kendra-role`** in the **Search** box above **Role name**.
3. From the suggested options, click on `kendra-role`.
4. In **Summary**, choose **Attach policies**.

5. In **Attach permissions**, in the **Search** box, enter **S3** and select the checkbox next to the **AmazonS3ReadOnlyAccess** policy from the suggested options.
6. Choose **Attach policy**. On the **Summary** page, you will now see two policies attached to the IAM role.
7. Return to the Amazon Kendra console at https://console.aws.amazon.com/kendra/ and wait for the status of your index to change from **Creating** to **Active** before continuing to the next step.

To update your IAM role (AWS CLI)

1. Save the following text in a JSON file called `kendra-S3-access-policy.json` in a text editor on your local device.

   Replace amzn-s3-demo-bucket with your S3 bucket name, *`aws-region`* with your AWS region, *`aws-account-id`* with your 12-digit AWS account ID, and *`kendra-index-id`* with your saved `kendra-index-id`.
2. To create an IAM policy to access your S3 bucket, use the create-policy command:

   Linux

   ```
   aws iam create-policy \
             --policy-name kendra-S3-access-policy \
             --policy-document file://path/kendra-S3-access-policy.json
   ```

   Where:

   - *`path/`* is the filepath to `kendra-S3-access-policy.json` on your local device.

   macOS

   ```
   aws iam create-policy \
             --policy-name kendra-S3-access-policy \
             --policy-document file://path/kendra-S3-access-policy.json
   ```

   Where:

   - *`path/`* is the filepath to `kendra-S3-access-policy.json` on your local device.

Windows

```
aws iam create-policy ^
          --policy-name kendra-S3-access-policy ^
          --policy-document file://path/kendra-S3-access-policy.json
```

Where:

- *path/* is the filepath to `kendra-S3-access-policy.json` on your local device.

3. Copy the Amazon Resource Name (ARN) to your text editor and save it locally as `kendra-S3-access-arn`.

> **Note**
>
> The ARN has a format similar to *arn:aws:iam::123456789012:role/kendra-S3-access-policy*. You need the ARN you saved as `kendra-S3-access-arn` to attach the `kendra-S3-access-policy` to your IAM role.

4. To attach the `kendra-S3-access-policy` to your Amazon Kendra IAM role, use the [attach-role-policy] command:

Linux

```
aws iam attach-role-policy \
          --policy-arn policy-arn \
          --role-name kendra-role
```

Where:

- *policy-arn* is your saved `kendra-S3-access-arn`.

macOS

```
aws iam attach-role-policy \
          --policy-arn policy-arn \
          --role-name kendra-role
```

Where:

- *policy-arn* is your saved `kendra-S3-access-arn`.

Windows

```
aws iam attach-role-policy ^
          --policy-arn policy-arn ^
          --role-name kendra-role
```

Where:

- *policy-arn* is your saved `kendra-S3-access-arn`.

Creating Amazon Kendra custom search index fields

To prepare Amazon Kendra to recognize your metadata as custom document attributes, you create custom fields corresponding to Amazon Comprehend entity types. You input the following nine Amazon Comprehend entity types as custom fields:

- COMMERCIAL_ITEM
- DATE
- EVENT
- LOCATION
- ORGANIZATION
- OTHER
- PERSON
- QUANTITY
- TITLE

Important

Misspelled entity types will not be recognized by the index.

To create custom fields for your Amazon Kendra index (Console)

1. Open the Amazon Kendra console at https://console.aws.amazon.com/kendra/.
2. From the **Indexes** list, click on `kendra-index`.
3. From the left navigation panel, under **Data management**, choose **Facet definition**.
4. From the **Index fields** menu, choose **Add field**.
5. In the **Add index field** dialog box, do the following:
   a. In **Field name**, enter **`COMMERCIAL_ITEM`**.
   b. In **Data type**, choose **String list**.
   c. In **Usage types**, select **Facetable**, **Searchable**, and **Displayable**, and then choose **Add**.
   d. Repeat steps a to c for each Amazon Comprehend entity type: COMMERCIAL_ITEM, DATE, EVENT, LOCATION, ORGANIZATION, OTHER, PERSON, QUANTITY, TITLE.

The console displays successful field addition messages. You can choose to close them before you proceed with the next step.

To create custom fields for your Amazon Kendra index (AWS CLI)

1. Save the following text as a JSON file called `custom-attributes.json` in a text editor on your local device.

```
[
   {
       "Name": "COMMERCIAL_ITEM",
       "Type": "STRING_LIST_VALUE",
       "Search": {
           "Facetable": true,
           "Searchable": true,
           "Displayable": true
       }
   },
   {
       "Name": "DATE",
       "Type": "STRING_LIST_VALUE",
       "Search": {
           "Facetable": true,
           "Searchable": true,
           "Displayable": true
```

```
        }
    },
    {
        "Name": "EVENT",
        "Type": "STRING_LIST_VALUE",
        "Search": {
            "Facetable": true,
            "Searchable": true,
            "Displayable": true
        }
    },
    {
        "Name": "LOCATION",
        "Type": "STRING_LIST_VALUE",
        "Search": {
            "Facetable": true,
            "Searchable": true,
            "Displayable": true
        }
    },
    {
        "Name": "ORGANIZATION",
        "Type": "STRING_LIST_VALUE",
        "Search": {
            "Facetable": true,
            "Searchable": true,
            "Displayable": true
        }
    },
    {
        "Name": "OTHER",
        "Type": "STRING_LIST_VALUE",
        "Search": {
            "Facetable": true,
            "Searchable": true,
            "Displayable": true
        }
    },
    {
        "Name": "PERSON",
        "Type": "STRING_LIST_VALUE",
        "Search": {
            "Facetable": true,
            "Searchable": true,
```

```
            "Displayable": true
        }
    },
    {
        "Name": "QUANTITY",
        "Type": "STRING_LIST_VALUE",
        "Search": {
            "Facetable": true,
            "Searchable": true,
            "Displayable": true
        }
    },
    {
        "Name": "TITLE",
        "Type": "STRING_LIST_VALUE",
        "Search": {
            "Facetable": true,
            "Searchable": true,
            "Displayable": true
        }
    }
]
```

2. To create custom fields in your index, use the update-index command:

   Linux

   ```
   aws kendra update-index \
           --id kendra-index-id \
           --document-metadata-configuration-updates file://path/custom-
   attributes.json \
           --region aws-region
   ```

   Where:

   - *kendra-index-id* is your saved `kendra-index-id`,
   - *path/* is the filepath to `custom-attributes.json` on your local device,
   - *aws-region* is your AWS region.

macOS

```
aws kendra update-index \
        --id kendra-index-id \
        --document-metadata-configuration-updates file://path/custom-
attributes.json \
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *path/* is the filepath to `custom-attributes.json` on your local device,
- *aws-region* is your AWS region.

Windows

```
aws kendra update-index ^
        --id kendra-index-id ^
        --document-metadata-configuration-updates file://path/custom-
attributes.json ^
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *path/* is the filepath to `custom-attributes.json` on your local device,
- *aws-region* is your AWS region.

3. To verify that the custom attributes have been added to your index, use the describe-index command:

Linux

```
aws kendra describe-index \
        --id kendra-index-id \
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

macOS

```
aws kendra describe-index \
        --id kendra-index-id \
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

Windows

```
aws kendra describe-index ^
        --id kendra-index-id ^
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

Adding the Amazon S3 bucket as a data source for the index

Before you can sync your index, you must connect your S3 data source to it.

To connect an S3 bucket to your Amazon Kendra index (Console)

1. Open the Amazon Kendra console at https://console.aws.amazon.com/kendra/.
2. From the **Indexes** list, click on `kendra-index`.
3. From the left navigation menu, under **Data management**, choose **Data sources**.

4. Under the **Select data source connector type** section, navigate to **Amazon S3**, and choose **Add connector**.
5. In the **Specify data source details** page, do the following:
    a. Under **Name and description**, for **Data source name**, enter **`S3-data-source`**.
    b. Keep the **Description** section blank.
    c. Keep the default settings for **Tags**.
    d. Choose **Next**.
6. On the **Configure sync settings** page, in the **Sync scope** section, do the following:
    a. In **Enter the data source location**, choose **Browse S3**.
    b. In **Choose resources**, select your S3 bucket and then choose **Choose**.
    c. In **Metadata files prefix folder location**, choose **Browse S3**.
    d. In **Choose resources**, click on the name of your bucket from the list of buckets.
    e. For **Objects**, select the option box for `metadata` and choose **Choose**. The location field should now say `metadata/`.
    f. Keep the default settings for **Access control list configuration file location**, **Select decryption key**, and **Additional configuration**.
7. For **IAM role**, on the **Configure sync settings** page, choose `kendra-role`.
8. On the **Configure sync settings** page, under **Sync run schedule**, for **Frequency**, choose **Run on demand** and then choose **Next**.
9. On the **Review and create** page, review your choices for the data source details and choose **Add data source**.

To connect an S3 bucket to your Amazon Kendra index (AWS CLI)

1. Save the following text as a JSON file called `S3-data-connector.json` in a text editor on your local device.

```
{
   "S3Configuration":{
      "BucketName":"amzn-s3-demo-bucket",
      "DocumentsMetadataConfiguration":{
         "S3Prefix":"metadata"
      }
   }
```

```
}
```

Replace amzn-s3-demo-bucket with the name of your S3 bucket.

2. To connect your S3 bucket to your index, use the [create-data-source](https://docs.aws.amazon.com/cli/latest/reference/kendra/create-data-source.html) command:

Linux

```
aws kendra create-data-source \
        --index-id kendra-index-id \
        --name S3-data-source \
        --type S3 \
        --configuration file://path/S3-data-connector.json \
        --role-arn role-arn \
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *path/* is the filepath to `S3-data-connector.json` on your local device,
- *role-arn* is your saved `kendra-role-arn`,
- *aws-region* is your AWS region.

macOS

```
aws kendra create-data-source \
        --index-id kendra-index-id \
        --name S3-data-source \
        --type S3 \
        --configuration file://path/S3-data-connector.json \
        --role-arn role-arn \
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *path/* is the filepath to `S3-data-connector.json` on your local device,
- *role-arn* is your saved `kendra-role-arn`,
- *aws-region* is your AWS region.

Windows

```
aws kendra create-data-source ^
        --index-id kendra-index-id ^
        --name S3-data-source ^
        --type S3 ^
        --configuration file://path/S3-data-connector.json ^
        --role-arn role-arn ^
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *path/* is the filepath to `S3-data-connector.json` on your local device,
- *role-arn* is your saved `kendra-role-arn`,
- *aws-region* is your AWS region.

3. Copy the connector `Id` and save it in a text editor as `S3-connector-id`. The `Id` helps you track the status of the data-connection process.

4. To ensure that your S3 data source was connected successfully, use the describe-data-source command:

Linux

```
aws kendra describe-data-source \
        --id S3-connector-id \
        --index-id kendra-index-id \
        --region aws-region
```

Where:

- *S3-connector-id* is your saved `S3-connector-id`,
- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

macOS

```
aws kendra describe-data-source \
        --id S3-connector-id \
        --index-id kendra-index-id \
        --region aws-region
```

Where:

- *S3-connector-id* is your saved `S3-connector-id`,
- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

Windows

```
aws kendra describe-data-source ^
        --id S3-connector-id ^
        --index-id kendra-index-id ^
        --region aws-region
```

Where:

- *S3-connector-id* is your saved `S3-connector-id`,
- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

At the end of this step, your Amazon S3 data source is connected to the index.

Syncing the Amazon Kendra index

With the Amazon S3 data source added, you now sync your Amazon Kendra index to it.

To sync your Amazon Kendra index (Console)

1. Open the Amazon Kendra console at https://console.aws.amazon.com/kendra/.
2. From the **Indexes** list, click on `kendra-index`.

3. From the left navigation menu, choose **Data sources**.
4. From **Data sources**, select `S3-data-source`.
5. From the top navigation bar, choose **Sync now**.

To sync your Amazon Kendra index (AWS CLI)

1. To sync your index, use the start-data-source-sync-job command:

   Linux

   ```
   aws kendra start-data-source-sync-job \
           --id S3-connector-id \
           --index-id kendra-index-id \
           --region aws-region
   ```

   Where:

   - *`S3-connector-id`* is your saved `S3-connector-id`,
   - *`kendra-index-id`* is your saved `kendra-index-id`,
   - *`aws-region`* is your AWS region.

   macOS

   ```
   aws kendra start-data-source-sync-job \
           --id S3-connector-id \
           --index-id kendra-index-id \
           --region aws-region
   ```

   Where:

   - *`S3-connector-id`* is your saved `S3-connector-id`,
   - *`kendra-index-id`* is your saved `kendra-index-id`,
   - *`aws-region`* is your AWS region.

   Windows

   ```
   aws kendra start-data-source-sync-job ^
   ```

```
        --id S3-connector-id ^
        --index-id kendra-index-id ^
        --region aws-region
```

Where:

- *S3-connector-id* is your saved S3-connector-id,
- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

2. To check the status of the index sync, use the [list-data-source-sync-jobs](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/kendra/list-data-source-sync-jobs.html) command:

   Linux

```
aws kendra list-data-source-sync-jobs \
        --id S3-connector-id \
        --index-id kendra-index-id \
        --region aws-region
```

   Where:

   - *S3-connector-id* is your saved S3-connector-id,
   - *kendra-index-id* is your saved kendra-index-id,
   - *aws-region* is your AWS region.

   macOS

```
aws kendra list-data-source-sync-jobs \
        --id S3-connector-id \
        --index-id kendra-index-id \
        --region aws-region
```

   Where:

   - *S3-connector-id* is your saved S3-connector-id,
   - *kendra-index-id* is your saved kendra-index-id,
   - *aws-region* is your AWS region.

Windows

```
aws kendra list-data-source-sync-jobs ^
        --id S3-connector-id ^
        --index-id kendra-index-id ^
        --region aws-region
```

Where:

- *S3-connector-id* is your saved `S3-connector-id`,
- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

At the end of this step, you have created a searchable and filterable Amazon Kendra index for your dataset.

Step 5: Querying the Amazon Kendra index

Your Amazon Kendra index is now ready for natural language queries. When you search your index, Amazon Kendra uses all the data and metadata you provided to return the most accurate answers to your search query.

There are three kinds of queries that Amazon Kendra can answer:

- Factoid queries ("who", "what", "when", or "where" questions)
- Descriptive queries ("how" questions)
- Keyword searches (questions whose intent and scope are not clear)

Topics

Querying your Amazon Kendra index

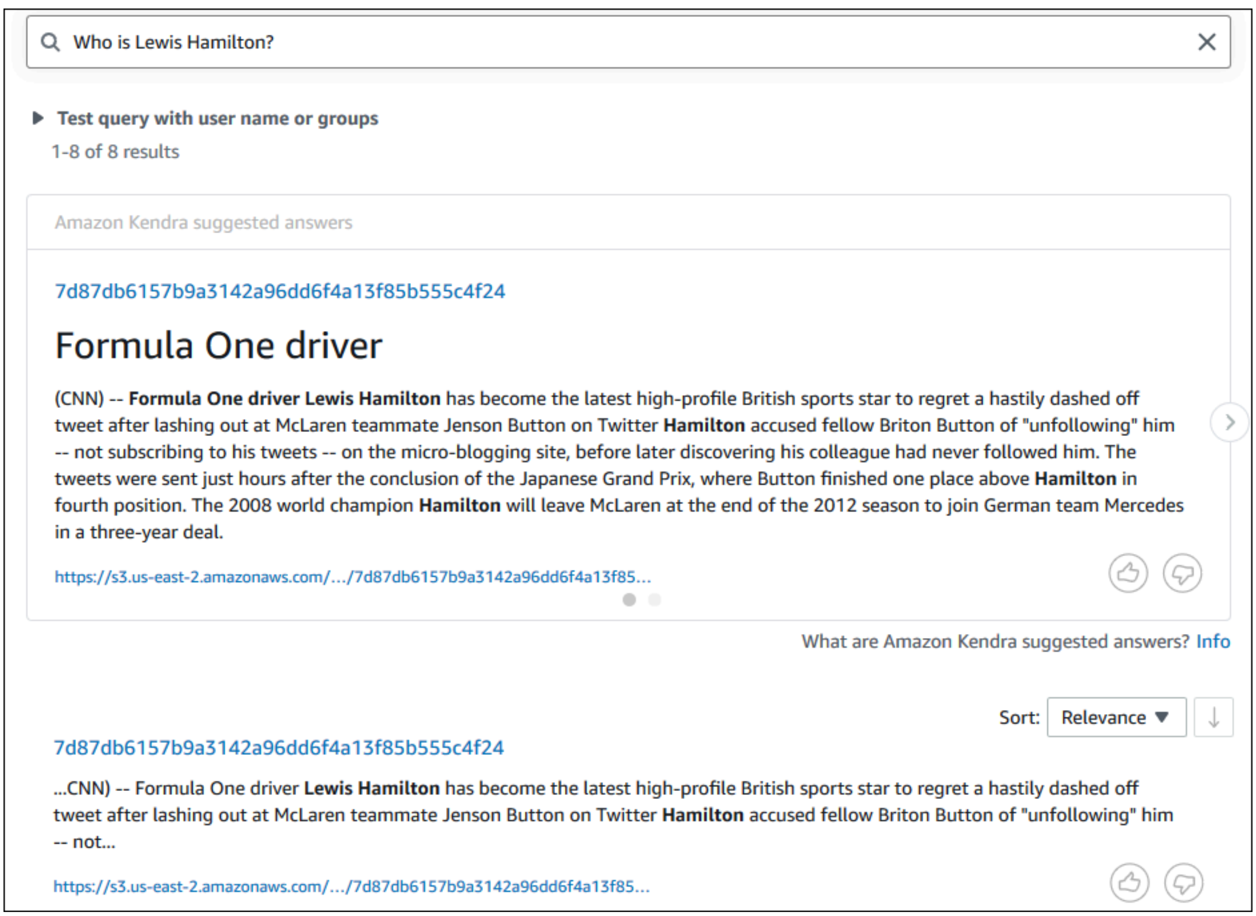
You can query your Amazon Kendra index using questions that correspond to the three kinds of queries that Amazon Kendra supports. For more information, see Queries.

The example questions in this section have been chosen based on the sample dataset.

To query your Amazon Kendra index (Console)

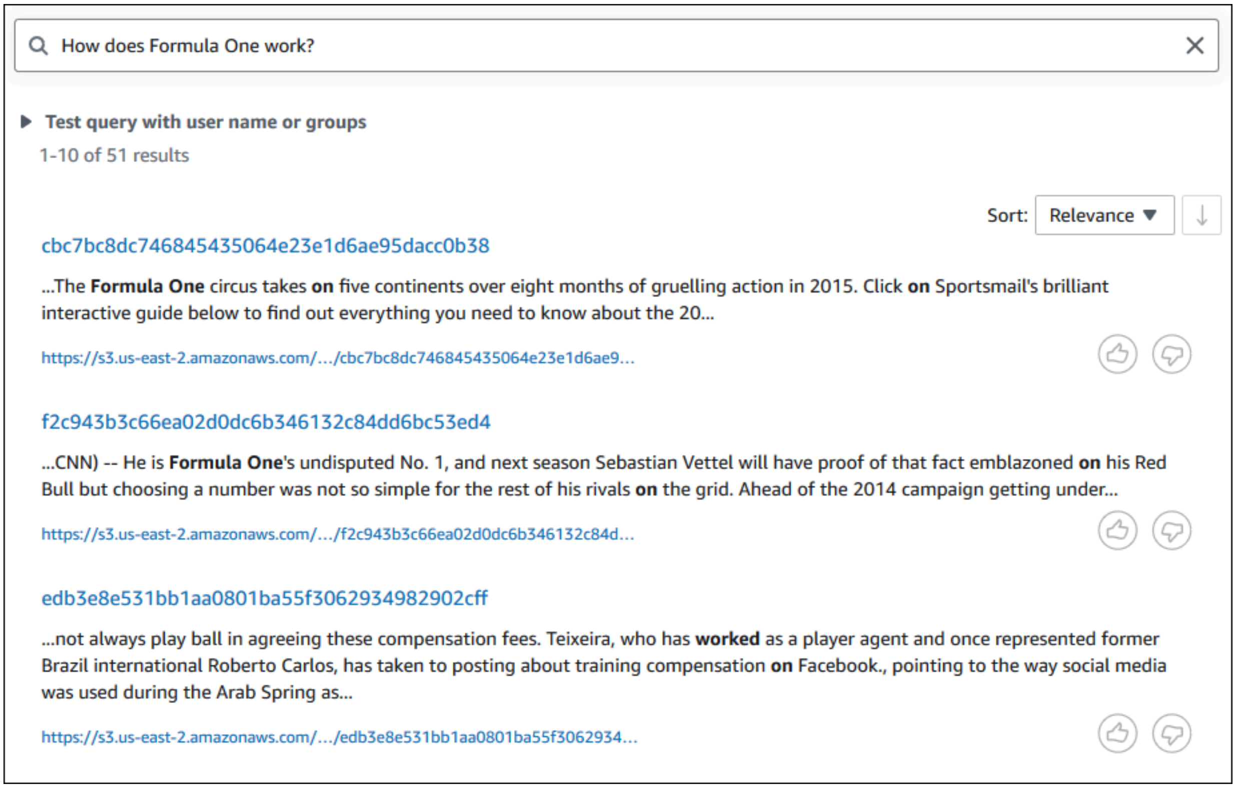
1. Open the Amazon Kendra console at https://console.aws.amazon.com/kendra/.
2. From the **Indexes** list, click on `kendra-index`.
3. From the left navigation menu, choose the option to search your index.
4. To run a sample factoid query, enter **`Who is Lewis Hamilton?`** in the search box and press enter.

   The first returned result is the Amazon Kendra suggested answer, together with the data file containing the answer. The rest of the results form the set of recommended documents.



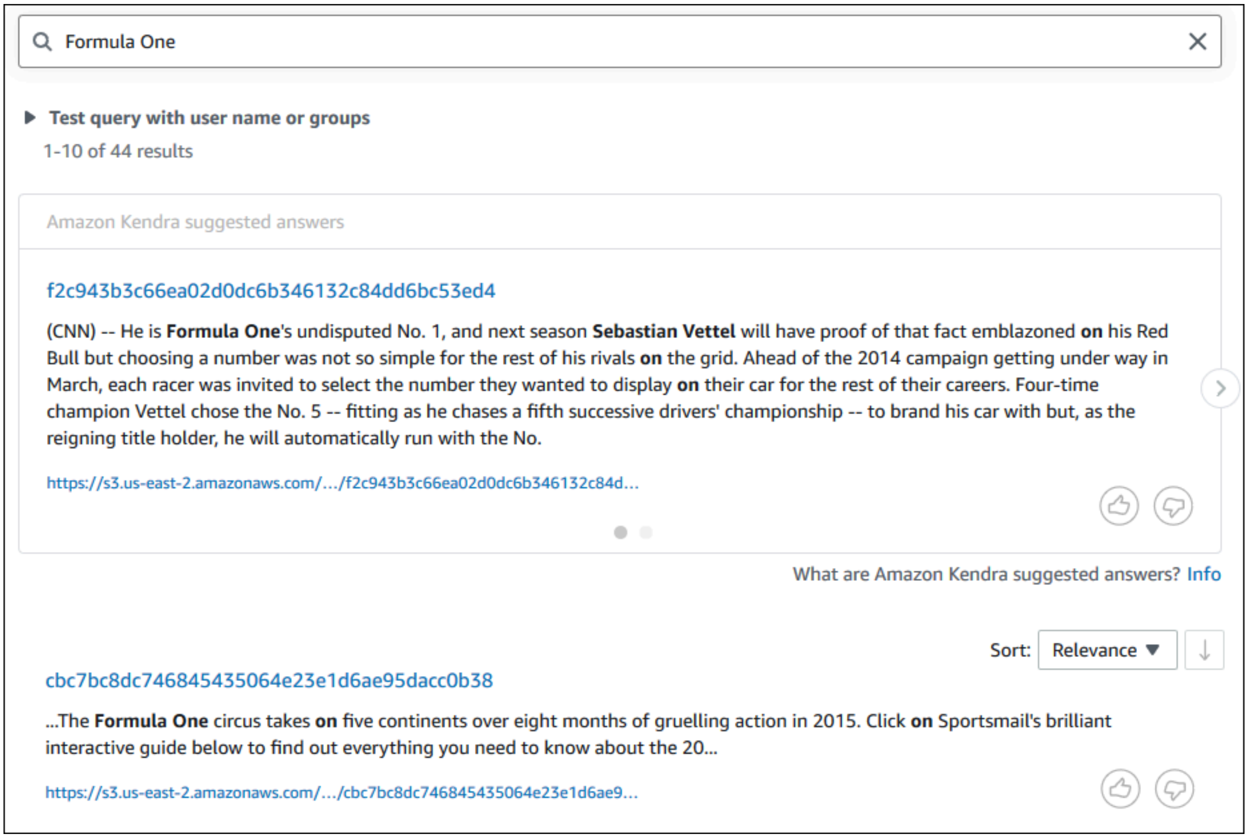
5. To run a descriptive query, enter **How does Formula One work?** in the search box and press enter.

   You will see another result returned by the Amazon Kendra console, this time with the relevant phrase highlighted.



6. To run a keyword search, enter **Formula One** in the search box and press enter.

   You will see another result returned by the Amazon Kendra console, followed by the results for all other mentions of the phrase in the dataset.



To query your Amazon Kendra index (AWS CLI)

1. To run a sample factoid query, use the query command:

   Linux

   ```
   aws kendra query \
           --index-id kendra-index-id \
           --query-text "Who is Lewis Hamilton?" \
           --region aws-region
   ```

   Where:

   - *kendra-index-id* is your saved `kendra-index-id`,
   - *aws-region* is your AWS region.

macOS

```
aws kendra query \
        --index-id kendra-index-id \
        --query-text "Who is Lewis Hamilton?" \
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

Windows

```
aws kendra query ^
        --index-id kendra-index-id ^
        --query-text "Who is Lewis Hamilton?" ^
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

The AWS CLI displays the results of your query.

2. To run a sample descriptive query, use the query command:

Linux

```
aws kendra query \
        --index-id kendra-index-id \
        --query-text "How does Formula One work?" \
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,

- *aws-region* is your AWS region.

macOS

```
aws kendra query \
        --index-id kendra-index-id \
        --query-text "How does Formula One work?" \
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

Windows

```
aws kendra query ^
        --index-id kendra-index-id ^
        --query-text "How does Formula One work?" ^
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

The AWS CLI displays the results to your query.

3. To run a sample keyword search, use the [query](https://docs.aws.amazon.com/cli/latest/reference/kendra/query.html) command:

Linux

```
aws kendra query \
        --index-id kendra-index-id \
        --query-text "Formula One" \
        --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

macOS

```
aws kendra query \
        --index-id kendra-index-id \
        --query-text "Formula One" \
        --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

Windows

```
aws kendra query ^
        --index-id kendra-index-id ^
        --query-text "Formula One" ^
        --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

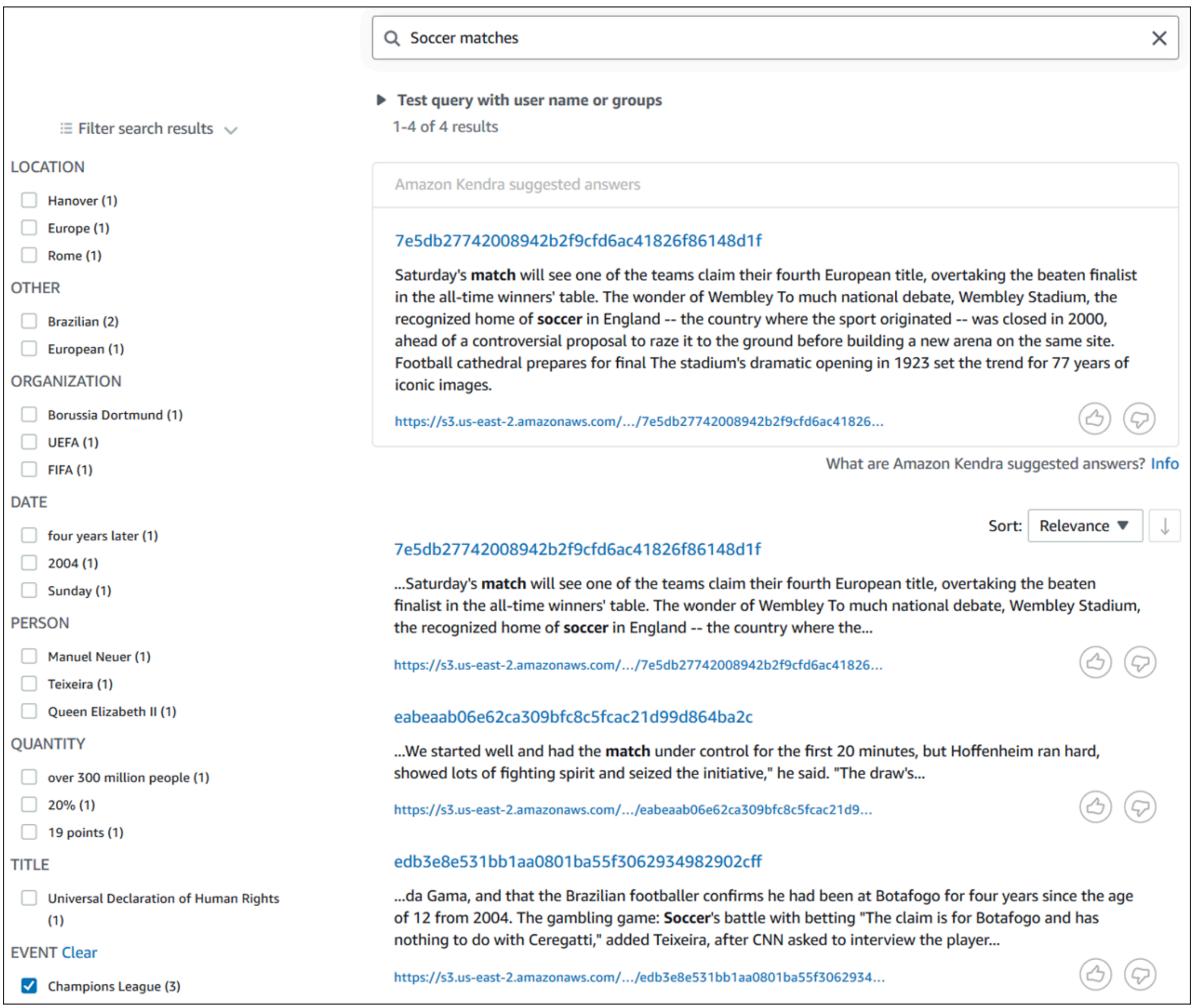
The AWS CLI displays the returned answers to your query.

Filtering your search results

You can filter and sort your search results using custom document attributes in the Amazon Kendra console. For more information on how Amazon Kendra processes queries, see Filtering queries.

To filter your search results (Console)

1. Open the Amazon Kendra console at https://console.aws.amazon.com/kendra/.
2. From the **Indexes** list, click on `kendra-index`.
3. From the left navigation menu, choose the option to search your index.
4. In the search box, enter **`Soccer matches`** as a query and press enter.
5. From the left navigation menu, choose **Filter search results** to see a list of facets you can use to filter your search.
6. Select the check box for "Champions League" under the **EVENT** subheading, to see your search results filtered only by the results containing "Champions League".



To filter your search results (AWS CLI)

1. To see the entities of a specific type (such as EVENT) that are available for a search, use the [query](https://docs.aws.amazon.com/kendra/latest/APIReference/API_Query.html) command:

   Linux

   ```
   aws kendra query \
           --index-id kendra-index-id \
           --query-text "Soccer matches" \
           --facets '[{"DocumentAttributeKey":"EVENT"}]' \
           --region aws-region
   ```

   Where:

   - *kendra-index-id* is your saved kendra-index-id,
   - *aws-region* is your AWS region.

   macOS

   ```
   aws kendra query \
           --index-id kendra-index-id \
           --query-text "Soccer matches" \
           --facets '[{"DocumentAttributeKey":"EVENT"}]' \
           --region aws-region
   ```

   Where:

   - *kendra-index-id* is your saved kendra-index-id,
   - *aws-region* is your AWS region.

   Windows

   ```
   aws kendra query ^
           --index-id kendra-index-id ^
           --query-text "Soccer matches" ^
           --facets '[{"DocumentAttributeKey":"EVENT"}]' ^
           --region aws-region
   ```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

The AWS CLI displays the search results. To get a list of facets of type EVENT, navigate to the "FacetResults" section of the AWS CLI output to see a list of filterable facets with their counts. For example, one of the facets is "Champions League".

> **Note**
>
> Instead of EVENT, you can choose any of the index fields you created in the section called "Creating an Amazon Kendra index" for the DocumentAttributeKey value.

2. To run the same search but filter only by the results containing "Champions League", use the query command:

   Linux

```
aws kendra query \
        --index-id kendra-index-id \
        --query-text "Soccer matches" \
        --attribute-filter '{"ContainsAny":{"Key":"EVENT","Value":
{"StringListValue":["Champions League"]}}}' \
        --region aws-region
```

   Where:

   - *kendra-index-id* is your saved kendra-index-id,
   - *aws-region* is your AWS region.

   macOS

```
aws kendra query \
        --index-id kendra-index-id \
        --query-text "Soccer matches" \
        --attribute-filter '{"ContainsAny":{"Key":"EVENT","Value":
{"StringListValue":["Champions League"]}}}' \
```

```
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

Windows

```
aws kendra query ^
        --index-id kendra-index-id ^
        --query-text "Soccer matches" ^
        --attribute-filter '{"ContainsAny":{"Key":"EVENT","Value":
{"StringListValue":["Champions League"]}}}' ^
        --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

The AWS CLI displays the filtered search results.

Step 6: Cleaning up

Cleaning up your files

To stop incurring charges in your AWS account after you complete this tutorial, you can take the following steps:

1. **Delete your Amazon S3 bucket**

   For information about deleting a bucket, see Deleting a bucket.

2. **Delete your Amazon Kendra index**

   For information about deleting an Amazon Kendra index, see Deleting an index.

3. **Delete `converter.py`**

- **For Console:** Go to AWS CloudShell, and make sure the region is set to your AWS region. After the bash shell has loaded, type the following command into the environment and press enter.

```
rm converter.py
```

- **For AWS CLI:** Run the following command on a terminal window.

  Linux

  ```
  rm file/converter.py
  ```

  Where:

  - *file/* is the filepath to `converter.py` on your local device.

  macOS

  ```
  rm file/converter.py
  ```

  Where:

  - *file/* is the filepath to `converter.py` on your local device.

  Windows

  ```
  rm file/converter.py
  ```

  Where:

  - *file/* is the filepath to `converter.py` on your local device.

Learn more

To learn more about integrating Amazon Kendra into your workflow, you can check out the following blogposts:

- Content metadata tagging for enhanced search
- Build an intelligent search solution with automated content enrichment

To learn more about Amazon Comprehend, you can look at the *Amazon Comprehend Developer Guide*.

Monitoring and logging for Amazon Kendra

Topics

Monitoring your index (console)

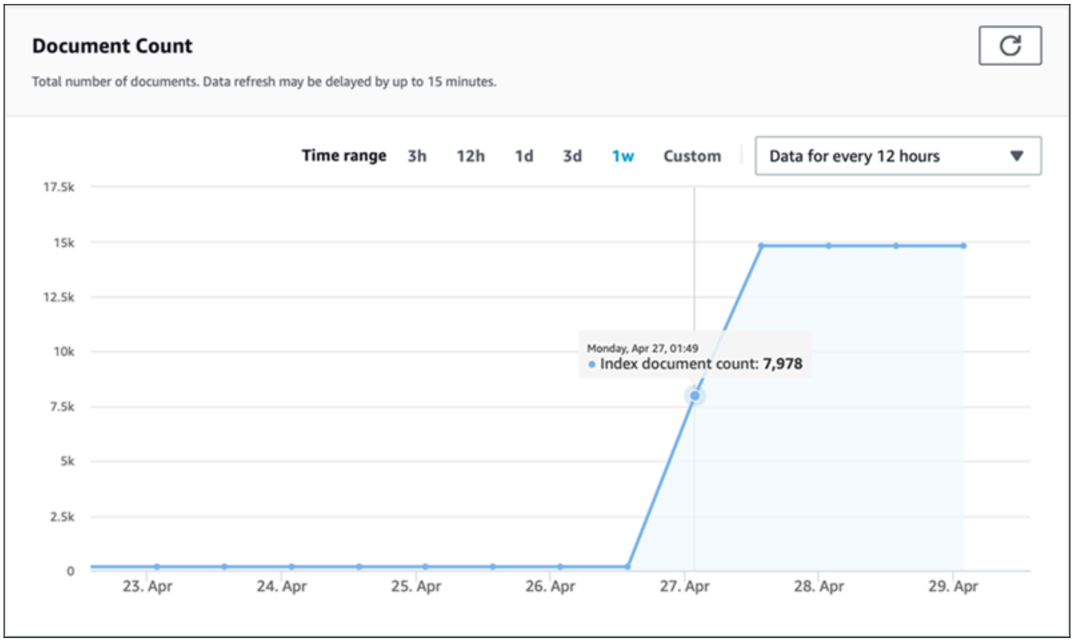
Use the Amazon Kendra console to monitor the state of indexes and data sources. You can use this information to track the size and storage requirements of your index and to monitor the progress and success of synchronization between your index and data sources.

To view index metrics (console)

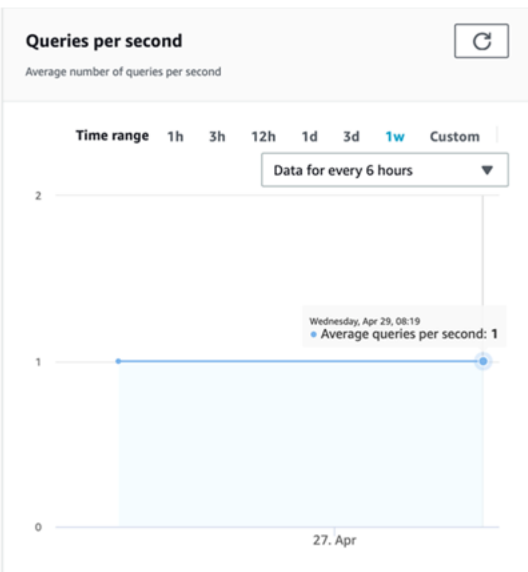
1. Sign into the AWS Management Console and open the Amazon Kendra console at https://console.aws.amazon.com/kendra/home.
2. From the list of indexes, choose the index to view.
3. Scroll the screen to see the index metrics.

You can see the following metrics about your index.

- **Document count—**The total number of documents indexed. This includes all documents from all data sources. Use this metric to determine if you need to purchase more or fewer storage units for your index.



- **Queries per second**—The number of index queries that are requested each second. Use this metric to determine if you need to purchase more or fewer query units for your index.



To monitor the progress and success of synchronization between your index and a data source, use the Amazon Kendra console. Use this information to help determine the health of your data source.

To view synchronization metrics (console)

1. Sign into the AWS Management Console and open the Amazon Kendra console at https://console.aws.amazon.com/kendra/home.

2. From the list of indexes, choose the index to view synchronization metrics for.

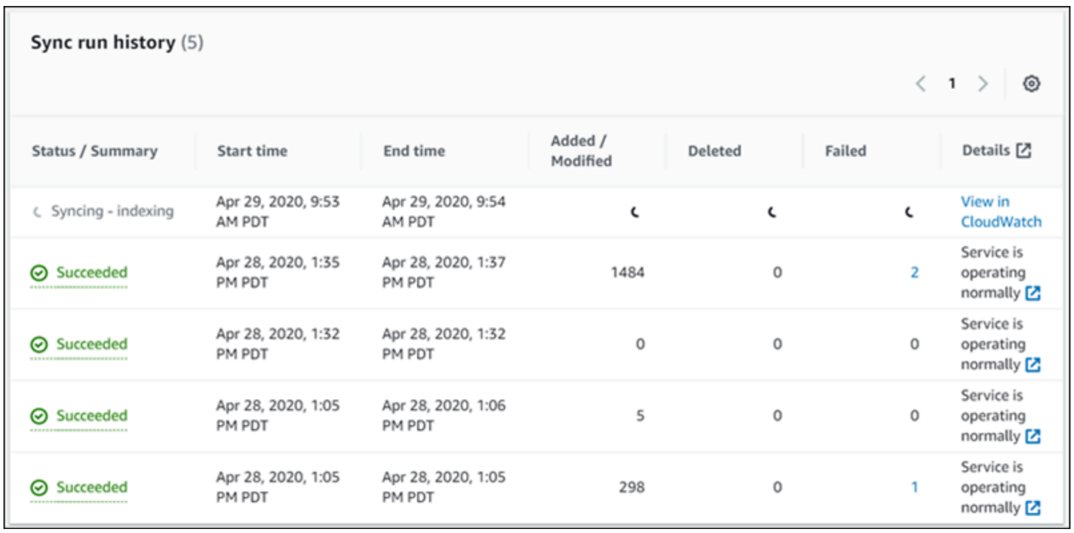
3. From the left menu, choose **Data sources**.

4. From the list of data sources, choose the data source to view.

5. Scroll the screen to see the sync run metrics.

You can see the following information.

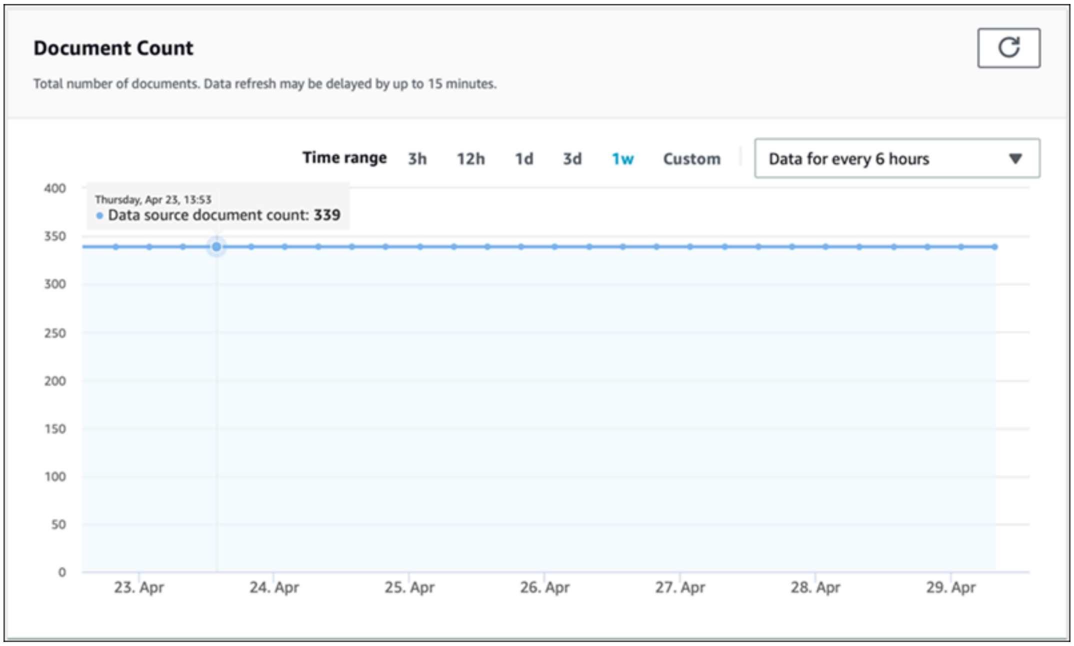
- **Sync run history**—Statistics about the synchronization run, including the start and end time, the number of documents added, deleted, and failed. If the sync run fails, there is a link to CloudWatch Logs with more information. Choose the settings icon in the upper left to change the columns that are displayed in the history. Use this information to determine the general health of your data source.



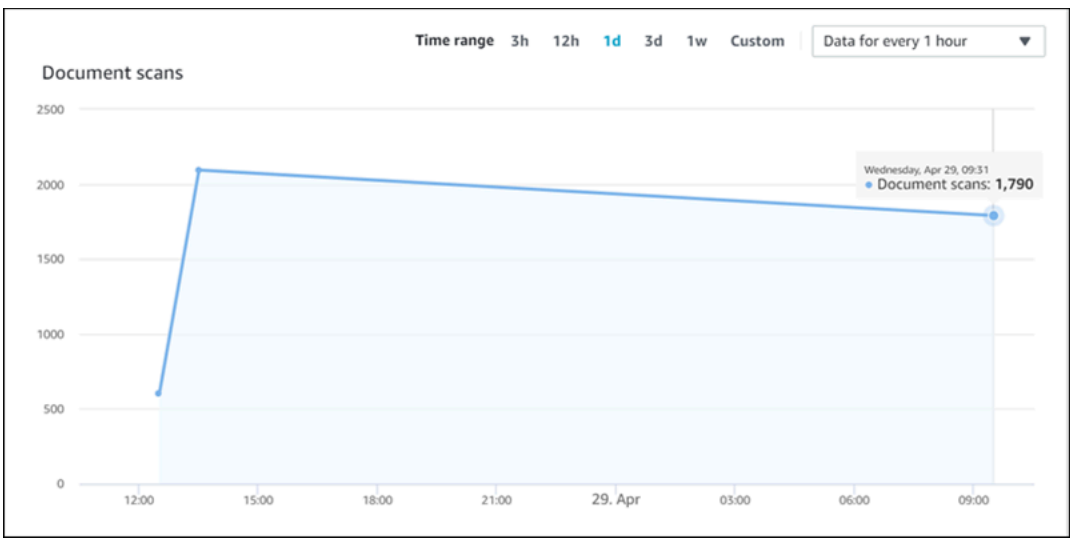
Sync run history (5)

Status / Summary	Start time	End time	Added / Modified	Deleted	Failed	Details
Syncing - indexing	Apr 29, 2020, 9:53 AM PDT	Apr 29, 2020, 9:54 AM PDT				View in CloudWatch
Succeeded	Apr 28, 2020, 1:35 PM PDT	Apr 28, 2020, 1:37 PM PDT	1484	0	2	Service is operating normally
Succeeded	Apr 28, 2020, 1:32 PM PDT	Apr 28, 2020, 1:32 PM PDT	0	0	0	Service is operating normally
Succeeded	Apr 28, 2020, 1:05 PM PDT	Apr 28, 2020, 1:06 PM PDT	5	0	0	Service is operating normally
Succeeded	Apr 28, 2020, 1:05 PM PDT	Apr 28, 2020, 1:05 PM PDT	298	0	1	Service is operating normally

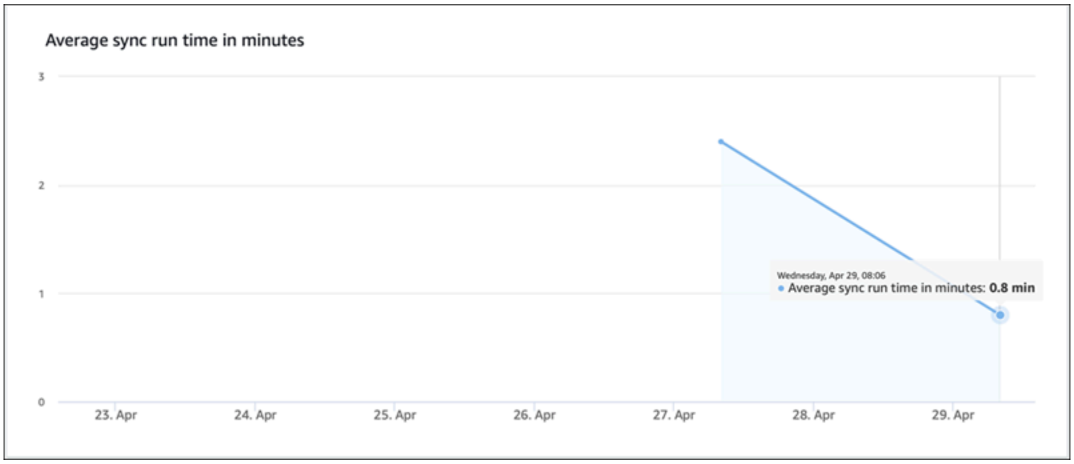
- **Document count**—The total number of documents indexed from this data source. This is the total of all documents added to the data source minus the total of all documents deleted from the data source. Use this information to determine how many documents from this data source are included in the index.



- **Document scans**—The total number of documents scanned during the sync run. This includes all documents in the data source, including those added, updated, deleted, or unchanged. Use this information to determine if Amazon Kendra is scanning all of the documents in the data source. The number of documents scanned affects the amount charged for the service.



- **Average sync run time in minutes**—The average length of time that it takes for a sync run to complete. The time that it takes to sync a data source affects the amount charged for the service.



Logging Amazon Kendra API calls with AWS CloudTrail logs

Amazon Kendra is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon Kendra. CloudTrail captures all API calls from Amazon Kendra as events, including calls from the Amazon Kendra console and from code calls to the Amazon Kendra APIs. If you create a trail, you can activate continuous delivery of CloudTrail events to and Amazon S3 bucket, including events for Amazon Kendra. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Amazon Kendra, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, including how to configure and activate it, see the AWS CloudTrail User Guide.

Amazon Kendra information in CloudTrail

CloudTrail is activated on your AWS account when you create the account. When activity occurs in Amazon Kendra, that activity is recorded in a CloudTrail event along with other AWS service events in the CloudTrail **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for Amazon Kendra, create a trail. A *trail* is a configuration that allows CloudTrail to deliver events as log files to a specified S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

CloudTrail logs all Amazon Kendra actions, which are documented in the API Reference. For example, calls to the `CreateIndex`, `CreateDataSource`, and `Query` operations generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. For more information, see the CloudTrail userIdentity Element.

Example: Amazon Kendra log file entries

A *trail* is a configuration that allows delivery of events as log files to a specified S3 bucket. CloudTrail log files contain one or more log entries. An *event* represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

Calls to the `Query` operation creates the following entry.

```
        {
            "eventVersion": "1.05",
            "userIdentity": {
                "type": "AssumedRole | FederatedUser | IAMUser | Root | SAMLUser | 
WebIdentityUser",
                "principalId": "principal ID",
                "arn": "ARN",
                "accountId": "account ID",
                "accessKeyId": "access key ID",
                "sessionContext": {
                    "sessionIssuer": {
                        "type": "Role",
                        "principalId": "principal Id",
                        "arn": "ARN",
```

```
                    "accountId": "account ID",
                    "userName": "user name"
                },
                "webIdFederationData": {

                },
                "attributes": {
                    "mfaAuthenticated": false,
                    "creationDate": "timestamp"
                }
            }
        },
        "eventTime": "timestamp",
        "eventSource": "kendra.amazonaws.com",
        "eventName": "Query",
        "awsRegion": "region",
        "sourceIPAddress": "source IP address",
        "userAgent": "user agent",
        "requestParameters": {
            "indexId": "index ID"
        },
        "responseElements": null,
        "requestID": "request ID",
        "eventID": "event ID",
        "eventType": "AwsApiCall",
        "recipientAccountId": "account ID"
    },
```

Logging Amazon Kendra Intelligent Ranking API calls with AWS CloudTrail logs

Amazon Kendra Intelligent Ranking is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon Kendra Intelligent Ranking. CloudTrail captures all API calls from Amazon Kendra intelligent Ranking as events, including code calls to the Amazon Kendra Intelligent Ranking APIs. If you create a trail, you can activate continuous delivery of CloudTrail events to and Amazon S3 bucket, including events for Amazon Kendra Intelligent Ranking. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Amazon Kendra Intelligent Ranking, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, including how to configure and activate it, see the AWS CloudTrail User Guide.

Amazon Kendra Intelligent Ranking information in CloudTrail

CloudTrail is activated on your AWS account when you create the account. When activity occurs in Amazon Kendra Intelligent Ranking, that activity is recorded in a CloudTrail event along with other AWS service events in the CloudTrail **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for Amazon Kendra Intelligent Ranking, create a trail. A *trail* is a configuration that allows CloudTrail to deliver events as log files to a specified S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

CloudTrail logs all Amazon Kendra Intelligent Ranking actions, which are documented in the API Reference. For example, calls to the `CreateRescoreExecutionPlan` generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. For more information, see the CloudTrail userIdentity Element.

Example: Amazon Kendra Intelligent Ranking log file entries

A *trail* is a configuration that allows delivery of events as log files to a specified S3 bucket. CloudTrail log files contain one or more log entries. An *event* represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

Calls to the `CreateRescoreExecutionPlan` operation creates the following entry.


```
{
            "eventVersion": "1.08",
            "userIdentity": {
                "type": "AssumedRole",
                "principalId": "principal ID",
                "arn": "ARN",
                "accountId": "account ID",
                "accessKeyId": "access key ID",
                "sessionContext": {
                    "sessionIssuer": {
                        "type": "Role",
                        "principalId": "principal ID",
                        "arn": "ARN",
                        "accountId": "account ID",
                        "userName": "user name"
                    },
                    "webIdFederationData": {},
                    "attributes": {
                        "creationDate": "yyyy-mm-ddThh:mm:ssZ",
                        "mfaAuthenticated": "false"
                    }
                }
            },
            "eventTime": "yyyy-mm-ddThh:mm:ssZ",
            "eventSource": "kendra-ranking.amazonaws.com",
            "eventName": "CreateRescoreExecutionPlan",
            "awsRegion": "region",
            "sourceIPAddress": "source IP address",
            "userAgent": "user agent",
            "requestParameters": {
                "name": "name",
                "description": "description",
                "clientToken": "client token"
            },
            "responseElements": {
                "id": "rescore execution plan ID",
                "arn": "rescore execution plan ARN"
            },
            "requestID": "request ID",
```

```
            "eventID": "event ID",
            "readOnly": false,
            "eventType": "AwsApiCall",
            "managementEvent": true,
            "recipientAccountId": "account ID",
            "eventCategory": "Management",
            "tlsDetails": {
                "tlsVersion": "TLS version",
                "cipherSuite": "cipher suite",
                "clientProvidedHostHeader": "kendra-ranking.[region].api.aws"
            }
        }
```

Monitoring Amazon Kendra with Amazon CloudWatch

To track the health of your indexes, use Amazon CloudWatch. With CloudWatch, you can get metrics for document synchronization for your index. You can also set up CloudWatch alarms to be notified when one or more metrics exceeds a threshold that you define. For example, you can monitor the number of documents submitted to be indexed or the number of documents that failed to be indexed.

You must have the appropriate CloudWatch permissions to monitor Amazon Kendra with CloudWatch. For more information, see Authentication and Access Control for Amazon CloudWatch in the *Amazon CloudWatch User Guide*.

Viewing Amazon Kendra metrics

View Amazon Kendra metrics using the CloudWatch console.

To view metrics (CloudWatch console)

1. Sign in to the AWS Management Console and open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. Choose **Metrics**, choose **All Metrics** and then choose **Kendra**.
3. Choose the dimension, choose a metric name, then choose **Add to graph**.
4. Choose a value for the date range. The metric count for the selected date range is displayed in the graph.

Creating an alarm

A CloudWatch alarm watches a single metric over a specified time period and performs one or more actions: sending a notification to an Amazon Simple Notification Service (Amazon SNS) top or Auto Scaling policy. The actions or actions are based on the value of the metric relative to a given threshold over a number of time periods that you specify. CloudWatch can also send you an Amazon SNS message when the alarm changes state.

CloudWatch alarms invoke actions only when the state changes and has persisted for the period that you specify.

To set an alarm

1. Sign in to the AWS Management Console and open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. Choose **Alarms** and then choose **Create alarm**.
3. Select a metric. Choose a **Kendra** metric for your index and data source. Also set the time as set number of hours, days, weeks, or custom.
4. Choose your statistic. For example, **Average**. Also choose your alarm trigger time period as a set number of minutes, hours, per day, or custom.
5. Choose your threshold to trigger the alarm, whether to use a static value or a band and the condition to meet for the threshold.
6. Choose the alarm state for the trigger, whether the metric must fall outside your set threshold, or another state. Select who/which email to send the alarm notification to.
7. If you are satisfied with the alarm, choose **Create alarm**.

> **Note**
>
> You must provide a name for your CloudWatch alarm.

CloudWatch Metrics for index synchronization Jobs

The following table describes the Amazon Kendra metrics for data source synchronization jobs.

If you use the API or CLI, you must specify the `Namespace` as 'AWS/Kendra' in addition to the `MetricName` of your choice when using GetMetricStatistics API.

Metric	Description
`DocumentsCrawled`	The number of documents that the synchronization job scanned or discovered during the run. Dimensions: • IndexId • DataSourceId Unit: Count
`DocumentsSubmittedForIndexing`	The number of documents that the synchronization job submitted to the index. Dimensions: • IndexId • DataSourceId Unit: Count
`DocumentsSubmittedForIndexingFailed`	The number of documents that failed indexing. Check the contents of the CloudWatch log for the synchronization job for details. Dimensions: • IndexId • DataSourceId Unit: Count
`DocumentsSubmittedForDeletion`	The number of documents that the synchronization job asked to be removed from the index.

Metric	Description
	Dimensions: • IndexId • DataSourceId Unit: Count
`DocumentsSubmittedForDeletionFailed`	The number of documents that failed to be deleted. Check the contents of the CloudWatch log for the synchronization job for details. Dimensions: • IndexId • DataSourceId Unit: Count

Metrics for Amazon Kendra data sources

The following table describes the Amazon Kendra metrics for data source synchronization jobs. Metrics marked with an asterisk (*) are used only for Amazon S3 data sources.

If you use the API or CLI, you must specify the `Namespace` as 'AWS/Kendra' in addition to the `MetricName` of your choice when using [GetMetricStatistics](https://docs.aws.amazon.com/AmazonCloudWatch/latest/APIReference/API_GetMetricStatistics.html) API.

Metric	Description
`DocumentsSkippedNoChange *`	The number of documents examined and found not to have changed so they weren't submitted for indexing. Dimensions: • IndexId • DataSourceId

Metric	Description
	Unit: Count
`DocumentsSkippedInvalidMetadata*`	The number of documents skipped because there was a problem with the associated metadata file. Check the contents of the CloudWatch log for the synchronization run for details. Dimensions: • IndexId • DataSourceId Unit: Count
`DocumentsCrawled`	The number of document files examined. Dimensions: • IndexId • DataSourceId Unit: Count
`DocumentsSubmittedForDeletion`	The number of documents examined that were deleted from the data source and submitted for deletion. Dimensions: • IndexId • DataSourceId Unit: Count

Metric	Description
`DocumentsSubmittedForDeletionFailed`	The number of documents that failed deletion from a data source. Dimensions: • IndexId • DataSourceId Unit: Count
`DocumentsSubmittedForIndexing`	The number of documents examined and submitted for indexing. Dimensions: • IndexId • DataSourceId Unit: Count
`DocumentsSubmittedForIndexingFailed`	The number of documents submitted for idexing that couldn't be indexed. Dimensions: • IndexId • DataSourceId Unit: Count

Metrics for indexed documents

The following table describes the Amazon Kendra metrics for indexed documents. For documents that are indexed using the BatchPutDocument operation, only the `IndexId` dimension is supported.

If you use the API or CLI, you must specify the `Namespace` as 'AWS/Kendra' in addition to the `MetricName` of your choice when using [GetMetricStatistics](#) API.

Metric	Description
`DocumentsIndexed`	The number of documents indexed. Dimensions: • IndexId • DataSourceId Unit: Count
`DocumentsFailedToIndex`	The number of documents that could not be indexed. Check the contents of the CloudWatc h log for details. Dimensions: • IndexId • DataSourceId Unit: Count
`IndexQueryCount`	The number of index queries per minute. Dimensions: • IndexId Unit: Count

Monitoring Amazon Kendra with Amazon CloudWatch Logs

Amazon Kendra uses Amazon CloudWatch Logs to give you insight into the operation of your data sources. Amazon Kendra logs process details for the documents as they are indexed. It logs errors from your data source that occur while your documents are being indexed. You use CloudWatch Logs to monitor, store and access the log files.

CloudWatch Logs stores log events in a log stream that is part of a log group. Amazon Kendra uses these features as follows:

- Log groups—Amazon Kendra stores all of your log streams in a single log group for each index. Amazon Kendra creates the log group when the index is created. The log group identifier always begins with "aws/kendra/".
- Log stream—Amazon Kendra creates a new data source log stream in the log group for each index synchronization job that you run. It also creates a new document log stream when a stream reaches approximately 500 entries.
- Log entries—Amazon Kendra creates a log entry in the log stream as it indexes documents. Each entry provides information about processing the document or any errors that are encountered.

For more information about using CloudWatch Logs, see [What Is Amazon Cloud Watch Logs] in the *Amazon Cloud Watch Logs User Guide*.

Amazon Kendra creates two types of log streams:

- Data source log streams
- Document log streams

Data source log streams

Data source log streams publish entries about your index synchronization jobs. Each synchronization job creates a new log stream that it uses to publish entries. The log stream name is:

```
data source id/YYYY-MM-DD-HH/data source sync job ID
```

A new log stream is created for each synchronization job run.

There are three types of log messages published to a data source log stream:

- A log message for a document that failed to be sent for indexing. The following is an example of this message for a document in an S3 data source:

```
{
    "DocumentId": "document ID",
    "S3Path": "s3://bucket/prefix/object",
    "Message": "Failed to ingest document via BatchPutDocument.",
    "ErrorCode": "InvalidRequest",
    "ErrorMessage": "No document metadata configuration found for document attribute
 key  city."
}
```

- A log message for a document that failed to be sent for deletion. The following is an example of this message:

```
{
    "DocumentId": "document ID",
    "Message": "Failed to delete document via BatchDeleteDocument.",
    "ErrorCode": "InvalidRequest",
    "ErrorMessage": "Document can't be deleted because it doesn't exist."
}
```

- A log message when an invalid metadata file for a document in an Amazon S3 bucket is found. The following is an example of this message.

```
{
    "Message": "Found invalid metadata
 file bucket/prefix/filename.extension.metadata.json."
}
```

- For SharePoint and database connectors, Amazon Kendra only writes messages to the log stream if a document can't be indexed. The following is an example of the error message that Amazon Kendra logs.

```
{
    "DocumentID": "document ID",
    "IndexID": "index ID",
    "SourceURI": "",
    "CrawlStatus": "FAILED",
    "ErrorCode": "403",
    "ErrorMessage": "Access Denied",
    "DataSourceErrorCode": "403"
}
```

Document log streams

Amazon Kendra logs information about processing documents while they are being indexed. It logs a set of messages for documents stored in an Amazon S3 data source. It logs errors only for documents stored in a Microsoft SharePoint or a database data source.

If the documents were added to the index using the BatchPutDocument operation, the log stream is named as follows:

```
YYYY-MM-DD-HH/UUID
```

If the documents were added to the index using a datasource, the log stream is named as follows:

```
dataSourceId/YYYY-MM-DD-HH/UUID
```

Each log stream contains up to 500 messages.

If indexing a document fails, this message is output to the log stream:

```
{
    "DocumentId": "document ID",
    "IndexName": "index name",
    "IndexId": "index ID"
    "SourceURI": "source URI"
    "IndexingStatus": "DocumentFailedToIndex",
    "ErrorCode": "400 | 500",
    "ErrorMessage": "message"
}
```

View Amazon Kendra metrics for your synchronization jobs

You can view a document-level sync run history report in CloudWatch for your data source sync job by selecting **View Report**. A sync run history report will have details about the progress and status of each document in the sync job. It shows if a document succeeded, failed, or was skipped during the crawl, sync, and index stages. You'll also find any error messages related to failed or skipped documents. If the report doesn't show results for an in-progress sync job, the logs may not be available yet. Check back later as data is emitted to the report as events occur during the sync process.

To access your sync run history report, take the following steps:

1. Open the Amazon Kendra console at https://console.aws.amazon.com/kendra/.
2. From the left navigation menu, under **Data management**, choose **Data sources**, and then choose your data source.
3. From your data source summary page, scroll down and select the **Sync history** tab.
4. From **Sync run history**, select **Actions**.
5. From **Actions**, select **View report**. You will be redirected to the CloudWatch console where you will be able to access your report.

> **Note**
>
> A sync run history records if a document was successfully indexed during ingestion, including attached ACLs and metadata, for all Amazon Kendra supported connectors.

If you're using the Amazon S3 connector:

In addition to the vieing the document-level sync run history report in CloudWatch, you can generate sync history reports for each document in your Amazon S3 data source and copy it to an Amazon S3 bucket. During this process, your data is encrypted using AWS KMS keys and can only be viewed by you. Reported document status can be one of the following: **Failed**, **Completed**, or **Succeeded with errors**. Before you can generate sync status reports for Amazon S3, you must do the following:

- Add the following Amazon Kendra service principal to your Amazon S3 access policy

```
{
    “Version”: “2012-10-17",
    “Statement”: [
        {
            “Sid”: “KendraS3Access”,
            “Effect”: “Allow”,
            “Principal”: {
                “Service”: “kendra.amazonaws.com”
            },
            “Action”: “s3:PutObject”,
            “Resource”: “arn:aws:s3:::your-manifest-bucket-name/*”
        }
    ]
```

```
}
```

- Create an Amazon S3 bucket with access permissions to Amazon Kendra

If you use the console, to generate a sync history report for Amazon S3, choose to activate the **Generate reports** option from **Sync history reports – *optional*** section on the **Data source details** page. Then, enter the Amazon S3 bucket location and choose from the configuration options available. Reports will be generated from the next sync after you have activated generate report.

If you delete the Amazon S3 bucket, you will lose your log data and will have to set up a new bucket to store new sync reports.

Note

A sync history report provides information only about whether an Amazon S3 connector successfully crawled and ingested data.

Security in Amazon Kendra

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud**—AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon Kendra, see AWS Services in Scope by Compliance Program.
- **Security in the cloud**—Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Amazon Kendra. The following topics show you how to configure Amazon Kendra to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon Kendra resources.

Topics

Data protection in Amazon Kendra

The AWS shared responsibility model applies to data protection in Amazon Kendra. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Amazon Kendra or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Encryption at rest

Amazon Kendra encrypts your data at rest with your choice of an encryption key. You can choose one of the following:

- An AWS-owned AWS KMS key. If you don't specify an encryption key your data is encrypted with this key by default.
- An AWS-managed KMS key in your account. This key is created, managed, and used on your behalf by Amazon Kendra. The key name is `aws/kendra`.
- A customer-managed key. You can provide the ARN of an encryption key that you created in your account. When you use a customer-managed KMS key, you must give the key a key policy that allows Amazon Kendra to use the key. Select a symmetric encryption customer-managed KMS key, Amazon Kendra does not support asymmetric KMS keys. For more information, see Key management.

Encryption in transit

Amazon Kendra uses the HTTPS protocol to communicate with your client application. It uses HTTPS and AWS signatures to communicate with other services on your application's behalf. If you use a VPC, you can use AWS PrivateLink to establish a private connection between your VPC and Amazon Kendra.

Key management

Amazon Kendra encrypts the contents of your index using one of three types of keys. You can choose one of the following:

- An AWS-owned AWS KMS. This is the default.
- An AWS-managed KMS key. This key is created in your account and is managed and used on your behalf by Amazon Kendra.
- A customer-managed KMS key. You can create the key when you are creating an Amazon Kendra index or data source, or you can create the key using the AWS KMS console. Select a symmetric encryption customer-managed KMS key. Amazon Kendra does not support asymmetric KMS keys. For more information, see Using Symmetric and Asymmetric Keys in the *AWS Key Management Service Developer Guide*.

Amazon Kendra Amazon Kendra Intelligent Ranking and interface VPC endpoints (AWS PrivateLink)

You can establish a private connection between your VPC and Amazon Kendra by creating an *interface VPC endpoint*. Interface endpoints are powered by AWS PrivateLink, a technology that allows you to privately access Amazon Kendra APIs without an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with Amazon Kendra APIs. Traffic between your VPC and Amazon Kendra doesn't leave the Amazon network.

Each interface endpoint is represented by one or more Elastic Network Interfaces in your subnets.

Considerations for Amazon Kendra and Amazon Kendra Intelligent Ranking VPC endpoints

Before you set up an interface VPC endpoint for Amazon Kendra or Amazon Kendra Intelligent Ranking, make sure that you review the prerequisites in the *Amazon VPC User Guide*.

Amazon Kendra and Amazon Kendra Intelligent Ranking supports making calls to all of its API actions from your VPC.

Creating an interface VPC endpoint for Amazon Kendra and Amazon Kendra Intelligent Ranking

You can create a VPC endpoint for the Amazon Kendra or Amazon Kendra Intelligent Ranking service using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI).

Create a VPC endpoint for Amazon Kendra using the following service name:

- com.amazonaws.*region*.kendra

Create a VPC endpoint for Amazon Kendra Intelligent Ranking using the following service name:

- aws.api.*region*.kendra-ranking

After you create a VPC endpoint, you can use the following example AWS CLI command that uses the `endpoint-url` parameter to specify an interface endpoint to the Amazon Kendra API:

```
aws kendra list-indices --endpoint-url https://VPC endpoint
```

`VPC endpoint` is the DNS name generated when the interface endpoint is created. This name includes the VPC endpoint ID, and the Amazon Kendra service name, which includes the region. For example, `vpce-1234-abcdef.kendra.us-west-2.vpce.amazonaws.com`.

If you activate private DNS for the endpoint, you can make API requests to Amazon Kendra using its default DNS name for the region. For example, `kendra.us-east-1.amazonaws.com`.

For more information, see Creating an interface endpoint in the *Amazon VPC User Guide*.

Creating a VPC endpoint policy for Amazon Kendra and Amazon Kendra Intelligent Ranking

You can attach an endpoint policy to your VPC endpoint that controls access to Amazon Kendra or Amazon Kendra Intelligent Ranking.

The policy for Amazon Kendra or Amazon Kendra Intelligent Ranking specifies the following information:

- The principal/authorized user that can perform actions.
- The actions that can be performed.
- The resources on which actions can be performed.

Example: VPC endpoint policy for Amazon Kendra actions

The following is an example of an endpoint policy for Amazon Kendra. When attached to an endpoint, this policy grants access to all available Amazon Kendra actions for all principals/authorized users on all resources.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "kendra:*"
         ],
```

```
            "Resource":"*"
        }
    ]
}
```

Example: VPC endpoint policy for Amazon Kendra Intelligent Ranking actions

The following is an example of an endpoint policy for Amazon Kendra Intelligent Ranking. When attached to an endpoint, this policy grants access to all available Amazon Kendra Intelligent Ranking actions for all principals/authorized users on all resources.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "kendra-ranking:*"
         ],
         "Resource":"*"
      }
   ]
}
```

For more information, see Controlling access to VPC endpoints using endpoint policies in the *Amazon VPC User Guide*.

Identity and access management for Amazon Kendra

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Amazon Kendra resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Amazon Kendra Identity-based policy examples
- AWS managed policies for Amazon Kendra
- Troubleshooting Amazon Kendra Identity and Access

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Amazon Kendra.

Service user – If you use the Amazon Kendra service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Amazon Kendra features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Amazon Kendra, see Troubleshooting Amazon Kendra Identity and Access.

Service administrator – If you're in charge of Amazon Kendra resources at your company, you probably have full access to Amazon Kendra. It's your job to determine which Amazon Kendra features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Amazon Kendra, see How Amazon Kendra works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Amazon Kendra. To view example Amazon Kendra identity-based policies that you can use in IAM, see Amazon Kendra Identity-based policy examples.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and AWS Multi-factor authentication in IAM in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

IAM Users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. To temporarily assume an IAM role in the AWS Management Console, you can switch from a user to an IAM role (console). You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Methods to assume a role in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Create a role for a third-party identity provider (federation) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the

principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Use an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – RCPs are JSON policies that you can use to set the maximum available permissions for resources in your accounts without updating the IAM policies attached to each resource that you own. The RCP limits permissions for resources in member accounts and can impact the effective permissions for identities, including the AWS account root user, regardless of whether they belong to your organization. For more information about Organizations and RCPs, including a list of AWS services that support RCPs, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How Amazon Kendra works with IAM

Before you use IAM to manage access to Amazon Kendra, you should understand what IAM features are available to use with Amazon Kendra. To get a high-level view of how Amazon Kendra and other AWS services work with IAM, see AWS Services That Work with IAM in the *IAM User Guide*.

Topics

Amazon Kendra identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. Amazon Kendra supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see IAM JSON Policy Elements Reference in the *IAM User Guide*.

Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

Policy actions in Amazon Kendra use the following prefix before the action: `kendra:`. For example, to grant someone permission to list Amazon Kendra indexes with the ListIndices API operation, you include the `kendra:ListIndices` action in their policy. Policy statements must include either an `Action` or `NotAction` element. Amazon Kendra defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": [
      "kendra:action1",
      "kendra:action2"
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action:

```
"Action": "kendra:Describe*"
```

To see a list of Amazon Kendra actions, see Actions Defined by Amazon Kendra in the *IAM User Guide*.

Resources

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

The Amazon Kendra index resource has the following ARN:

```
arn:${Partition}:kendra:${Region}:${Account}:index/${IndexId}
```

For more information about the format of ARNs, see Amazon Resource Names (ARNs) and AWS Service Namespaces.

For example, to specify an index in your statement, use the GUID of the index in the following ARN:

```
"Resource": "arn:aws:kendra:${Region}:${Account}:index/${GUID}"
```

To specify all indexes that belong to a specific account, use the wildcard (*):

```
"Resource": "arn:aws:${Region}:${Account}:index/*"
```

Some Amazon Kendra actions, such as those for creating resources, cannot be performed on a specific resource. In those cases, you must use the wildcard (*).

```
"Resource": "*"
```

To see a list of Amazon Kendra resource types and their ARNs, see Resources Defined by Amazon Kendra in the *IAM User Guide*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by Amazon Kendra.

Condition keys

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

Amazon Kendra does not provide any service-specific condition keys, but it does support using some global condition keys. To see all AWS global condition keys, see AWS Global Condition Context Keys in the *IAM User Guide*.

Examples

To view examples of Amazon Kendra identity-based policies, see Amazon Kendra Identity-based policy examples.

Amazon Kendra Resource-based policies

Amazon Kendra does not support resource-based policies.

Access control lists (ACLs)

Amazon Kendra does not support access control lists (ACLs) for access to AWS services and resources.

Authorization based on Amazon Kendra tags

You can associate tags with certain types of Amazon Kendra resources to authorize access to those resources. To control access based on tags, provide tag information in the condition element of a policy by using the `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

The following table lists the actions, corresponding resource types, and condition keys for tag-based access control. Each action is authorized based on the tags associated with the corresponding resource type.

Action	Resource type	Condition keys
CreateDataSource		`aws:RequestTag` , `aws:TagKeys`

Action	Resource type	Condition keys
CreateFaq		aws:RequestTag , aws:TagKeys
CreateIndex		aws:RequestTag , aws:TagKeys
API_ListTagsForResource	data source, FAQ, index	
TagResource	data source, FAQ, index	aws:RequestTag , aws:TagKeys
UntagResource	data source, FAQ, index	aws:TagKeys

For information about tagging Amazon Kendra resources, see Tags. For an example identity-based policy that limits access to a resource based on resource tags, see Tag-based policy examples. For more information about using tags to limit access to resources, see Controlling access using tags in the *IAM User Guide*.

Amazon Kendra IAM Roles

An IAM role is an entity within your AWS account that has specific permissions.

Using temporary credentials with Amazon Kendra

You can use temporary credentials to sign in with federation, assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as AssumeRole or GetFederationToken.

Amazon Kendra supports using temporary credentials.

Service roles

This feature allows a service to assume a service role on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in your IAM account and are owned by the account. This means that an IAM administrator can change the permissions for this role. However, doing so might break the functionality of the service.

Amazon Kendra supports service roles.

Choosing an IAM role in Amazon Kendra

When you create an index, call the `BatchPutDocument` operation, create a data source or create an FAQ, you must provide an access role Amazon Resource Name (ARN) that Amazon Kendra uses to access the required resources on your behalf. If you have previously created a role, then the Amazon Kendra console provides you with a list of roles to choose from. It's important to choose a role that allows access to the resources that you require. For more information, see IAM access roles for Amazon Kendra.

Amazon Kendra Identity-based policy examples

By default, users and roles don't have permission to create or modify Amazon Kendra resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see Creating Policies on the JSON Tab in the *IAM User Guide*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Amazon Kendra resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.

- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

AWS Managed (Predefined) Polices for Amazon Kendra

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. These policies are called AWS managed policies. AWS managed policies make it easier for you to assign permissions to users, groups, and roles than if you had to write the policies yourself. For more information, see Adding Permissions to a User in the *IAM User Guide*.

The following AWS managed policies, which you can attach to groups and roles in your account, are specific to Amazon Kendra:

- **AmazonKendraReadOnly** — Grants read-only access to Amazon Kendra resources.

- **AmazonKendraFullAccess** — Grants full access to create, read, update, delete, tag, and run all Amazon Kendra resources.

For the console, your role must also have `iam:CreateRole`, `iam:CreatePolicy`, `iam:AttachRolePolicy`, and `s3:ListBucket` permissions.

> **Note**
>
> You can review these permissions by signing in to the IAM console and searching for specific policies.

You can also create your own custom policies to allow permissions for Amazon Kendra API actions. You can attach these custom policies to the IAM roles or groups that require those permissions. For examples of IAM policies for Amazon Kendra, see [Amazon Kendra Identity-based policy examples](#).

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
```

```
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Accessing one Amazon Kendra index

In this example, you want to grant an user in your AWS account access to query an index.

Tag-based policy examples

Tag-based policies are JSON policy documents that specify the actions that a principal can perform on tagged resources.

Example: Use a tag to access a resource

This example policy grants a user or role in your AWS account permission to use the `Query` operation with any resource tagged with the key **`department`** and the value **`finance`**.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kendra:Query"
            ],
            "Resource": "*",
            "Condition": {
```

```
                "StringEquals": {
                    "aws:ResourceTag/department": "finance"
                }
            }
        }
    ]
}
```

Example: Use a tag to activate Amazon Kendra operations

This example policy grants a user or role in your AWS account permission to use any Amazon Kendra operation except `TagResource` operation with any resource tagged with the key **`department`** and the value **`finance`**.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kendra:*",
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "kendra:TagResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/department": "finance"
                }
            }
        }
    ]
}
```

Example: Use a tag to restrict access to an operation

This example policy restricts access for a user or role in your AWS account to use the `CreateIndex` operation unless the user provides the **`department`** tag and it has the allowed values **`finance`** and **`IT`**.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kendra:CreateIndex",
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": "kendra:CreateIndex",
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/department": "true"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": "kendra:CreateIndex",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringNotEquals": {
                    "aws:RequestTag/department": [
                        "finance",
                        "IT"
                    ]
                }
            }
        }
    ]
}
```

AWS managed policies for Amazon Kendra

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the *IAM User Guide*.

AWS managed policy: AmazonKendraReadOnly

Grants read-only access to Amazon Kendra resources. This policy includes the following permissions.

- `kendra` – Allows users to perform actions that return either a list of items or details about an item. This includes API operations that start with `Describe`, `List`, `Query`, `BatchGetDocumentStatus`, `GetQuerySuggestions`, or `GetSnapshots`.

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "kendra:Describe*",
                "kendra:List*",
                "kendra:Query",
                "kendra:BatchGetDocumentStatus",
                "kendra:GetQuerySuggestions",
                "kendra:GetSnapshots"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

AWS managed policy: AmazonKendraFullAccess

Grants full access to create, read, update, delete, tag, and run all Amazon Kendra resources. This policy includes the following permissions.

- `kendra`—Allows principals read and write access to all actions in the Amazon Kendra.
- `s3`—Allows principals get Amazon S3 bucket locations and list buckets.
- `iam`—Allows principals to pass and list roles.
- `kms`—Allows principals to describe and list AWS KMS keys and aliases.
- `secretsmanager`—Allows principals to create, describe, and list secrets.
- `ec2`—Allows principals to describe security groups, VCPs (Virtual Private Cloud), and subnets.
- `cloudwatch`—Allows principals to view Cloud Watch metrics.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "kendra.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:ListKeys",
                "kms:ListAliases",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:ListSecrets"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:DescribeSecret"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:AmazonKendra-*"
        },
        {
            "Effect": "Allow",
            "Action": "kendra:*",
            "Resource": "*"
        }
    ]
}
```

Amazon Kendra updates to AWS managed policies

View details about updates to AWS managed policies for Amazon Kendra since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Amazon Kendra Document history page.

Change	Description	Date
AmazonKendraReadOnly— Add permission to support GetSnapshots, BatchGetDocumentStatus APIs	Amazon Kendra added new APIs `GetSnapshots` and `BatchGetDocumentSt`	January 3, 2022

Change	Description	Date
	atus . GetSnapshots provides data that shows how your users interact with your search application. BatchGetDocumentSt atus monitors the progress of indexing your documents.	
AmazonKendraReadOnly— Add permission to support GetQuerySuggestions operation	Amazon Kendra added a new API GetQuerySuggestion s that allows access to get query suggestions for popular search queries, helping guide your users' search. When users type their search query, the suggested query helps autocomplete their search.	May 27, 2021
Amazon Kendra started tracking changes	Amazon Kendra started tracking changes for its AWS managed policies.	May 27, 2021

Troubleshooting Amazon Kendra Identity and Access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Amazon Kendra and IAM.

Topics

- I am not authorized to perform an action in Amazon Kendra
- I am not authorized to perform iam:PassRole
- I'm an administrator and I want to allow others to access Amazon Kendra
- I want to allow people outside of my AWS account to access my Amazon Kendra resources

I am not authorized to perform an action in Amazon Kendra

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your sign-in credentials.

The following example error occurs when the `mateojackson` user tries to use the console to view details about an index but does not have `kendra:`*`DescribeIndex`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 kendra:DescribeIndex on resource: index ARN
```

In this case, Mateo asks his administrator to update his policies to allow him to access the `index` resource using the `kendra:`*`DescribeIndex`* action.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Amazon Kendra.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Amazon Kendra. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I'm an administrator and I want to allow others to access Amazon Kendra

To allow others to access Amazon Kendra, you must grant permission to the people or applications that need access. If you are using AWS IAM Identity Center to manage people and applications, you assign permission sets to users or groups to define their level of access. Permission sets automatically create and assign IAM policies to IAM roles that are associated with the person or application. For more information, see Permission sets in the *AWS IAM Identity Center User Guide*.

If you are not using IAM Identity Center, you must create IAM entities (users or roles) for the people or applications that need access. You must then attach a policy to the entity that grants them the correct permissions in Amazon Kendra. After the permissions are granted, provide the credentials to the user or application developer. They will use those credentials to access AWS. To learn more about creating IAM users, groups, policies, and permissions, see IAM Identities and Policies and permissions in IAM in the *IAM User Guide*.

I want to allow people outside of my AWS account to access my Amazon Kendra resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Amazon Kendra supports these features, see How Amazon Kendra works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Security best practices

Amazon Kendra provides a number of security features to consider as you develop and implement your own security policies. The following best practices are general guidelines and don't represent a complete security solution. Because these best practices might not be appropriate or sufficient for your environment, treat them as helpful considerations rather than prescriptions.

Apply principle of least privilege

Amazon Kendra provides a granular access policy for applications using IAM roles. We recommend that the roles be granted only the minimum set of privileges required by the job, such as covering your application and access to log destination. We also recommend auditing the jobs for permissions on a regular basis and upon any change to your application.

Role-based access control (RBAC) permissions

Administrators should strictly control Role-based access control (RBAC) permissions for Amazon Kendra applications.

Logging and monitoring in Amazon Kendra

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Kendra applications. To monitor Amazon Kendra API calls, you can use AWS CloudTrail. To monitor the status of your jobs, use Amazon CloudWatch Logs.

- **Amazon CloudWatch Alarms**—Using CloudWatch alarms, you watch a single metric over a time period that you specify. If the metric exceeds a policy. CloudWatch alarms do not invoke actions when a metric is in a particular state. Rather the state must have changed and been maintained for a specified number of periods. For more information, see Monitoring Amazon Kendra with Amazon CloudWatch.
- **AWS CloudTrail Logs**—CloudTrail provides a record of actions taken by a user, role, or an AWS service in Amazon Kendra or Amazon Kendra Intelligent Ranking. Using the information collected by CloudTrail, you can determine the request that was made to Amazon Kendra, the IP address from which the request was made, who made the request, when it was made, and additional details. For more information, see Logging Amazon Kendra API calls with AWS CloudTrail logs and Logging Amazon Kendra Intelligent Ranking API calls with AWS CloudTrail logs.

Compliance validation for Amazon Kendra

Third-party auditors assess the security and compliance of Amazon Kendra as part of multiple Amazon Kendra compliance programs. Amazon Kendra is compliant with the following:

- Health Insurance Portability and Accountability Act (HIPAA)
- System and Organization Controls (SOC) 2
- Information Security Registered Assessors Program (IRAP)
- Federal Risk and Authorization Management Program (FedRAMP) Moderate in the US East/West regions
- Federal Risk and Authorization Management Program (FedRAMP) High in the AWS GovCloud (US-West) region

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using Amazon Kendra is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides—These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA Security and Compliance Whitepaper—This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS Compliance Resources—This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide*—The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub—This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in Amazon Kendra

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

With AWS global infrastructure, Amazon Kendra Enterprise Edition is fault tolerant, scalable, and highly available. Rolling back to previous versions of an index is not currently supported, but you can refresh or recreate portions of your index by deleting and adding existing data sources back into your index.

Infrastructure security in Amazon Kendra

As a managed service, Amazon Kendra is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well‐Architected Framework*.

You use AWS published API calls to access Amazon Kendra through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Configuration and vulnerability analysis in AWS Identity and Access Management

AWS handles basic security tasks like guest operating system (OS) and database patching, firewall configuration, and disaster recovery. These procedures have been reviewed and certified by the appropriate third parties. For more details, see the following resources:

- Shared Responsibility Model
- AWS: Overview of Security Processes (whitepaper)

The following resources also address configuration and vulnerability analysis in AWS Identity and Access Management (IAM):

- Compliance validation for AWS Identity and Access Management
- Security best practices and use cases in AWS Identity and Access Management.

Quotas for Amazon Kendra

Supported regions

For a list of AWS regions where Amazon Kendra is available, see Amazon Kendra regions and endpoints in the *Amazon Web Services General Reference*.

Important

Amazon Kendra GenAI Enterprise Edition indices are only available in the US East (N. Virginia), US West (Oregon), Europe (Ireland), Asia Pacific (Sydney), regions.

Quotas

Service quotas, also referred to as limits, are the maximum number of service resources for your AWS account. For more information, see Amazon Kendra service quotas in the *AWS General Reference*.

Note

Quotas for Amazon Kendra GenAI Enterprise Edition indices aren't currently available in the Service Quotas console.

Index quotas

Description	Default	Edition	Adjustable
Maximum number of indices per account	10	GenAI Enterprise, Developer, Enterprise	Yes
Amount of text extracted for an index in a single unit (Developer). You can't add extra units	3 GB	Developer	No

Description	Default	Edition	Adjustable
for extracting text for the Developer Edition.			
Amount of text extracted for an index in a single unit (Enterprise). You can add up to 100 extra units for extracting text for the Enterpris e Edition, or simply contact Support.	30 GB	GenAI Enterprise	Yes
Maximum number of storage capacity units per GenAI Enterprise Edition index	50	Enterprise	Yes

Data source connector quotas

Description	Default	Edition	Adjustable
Maximum number of data source connectors per index (Developer)	5	Developer	No
Maximum number of data source connectors per index (Enterprise)	50	Enterprise	Yes
Maximum number of data source	50	GenAI Enterprise	Yes

Description	Default	Edition	Adjustable
connectors per index (GenAI Enterprise)			
Maximum hours of connector sync usage per GenAI Enterprise Index per month	500	GenAI Enterprise	Yes
Maximum size of a single document or raw file that can be ingested when using a data source connector	50 MB	GenAI Enterprise, Developer, Enterprise	Yes
Maximum number of S3 prefixes in the access control list configuration file included in the Amazon S3 data source connector	100	Developer, Enterprise	No
Maximum size of the access control list configuration file included in the Amazon S3 data source connector	50 MB	Developer, Enterprise	Yes

FAQ quotas

Description	Default	Edition	Adjustable
Maximum number of FAQs per index	30	Developer, Enterprise	Yes
Maximum size of 1 FAQ	5 MB	Developer, Enterprise	Yes
Maximum number of results returned for FAQ	4	Developer, Enterprise	Yes
Maximum number of characters allowed for a FAQ question	300	Developer, Enterprise	No
Maximum number of characters in an FAQ answer	2000	Developer, Enterprise	No

Thesaurus quotas

Description	Default	Edition	Adjustable
Maximum number of thesauri per index	1	Developer, Enterprise	No
Maximum size of a thesaurus file	5 MB	Developer, Enterprise	Yes
Maximum number of synonym rules per thesaurus	10,000	Developer, Enterprise	Yes
Maximum number of synonyms per term	10	Developer, Enterprise	No

Description	Default	Edition	Adjustable
in all thesauri in an index			

Amazon Kendra experience quotas

Description	Default	Edition	Adjustable
Maximum number of Amazon Kendra experiences per index	50	Developer, Enterprise	Yes

Query and search results quotas

Note

Query per second is shared across retrieve and query API.

Description	Default	Edition	Adjustable
Amount of queries per second for an index in a single unit (Developer). You can't add extra units for queries for the Developer Edition.	0.05	Developer	No
Amount of queries per second for an index in a single unit (Enterprise). You can add up to 100	0.1	Enterprise	Yes

Description	Default	Edition	Adjustable
extra units for queries for the Enterprise Edition, or simply contact Support.			
Maximum number of query capacity units per GenAI Enterprise Edition index	100	GenAI Enterprise	Yes
Maximum number of characters per query text	1000	GenAI Enterprise, Developer, Enterprise	Yes
Maximum number of search results per query. Default is 100. To allow more than 100 results, simply contact Support.	100	Developer, Enterprise	Yes
Maximum number of search results per page	100	Developer, Enterprise	Yes
Maximum number of token words per query text before truncation. Default is 30. To allow more than 30 words, simply contact Support.	30	Developer, Enterprise	Yes

Description	Default	Edition	Adjustable
Maximum user-group list size per query attribute	1000	GenAI Enterprise, Developer, Enterprise	Yes
Maximum string list size per query attribute	10	GenAI Enterprise, Developer, Enterprise	Yes

Query suggestions quotas

Description	Default	Edition	Adjustable
Maximum number of query suggestions returned per [GetQuerySuggestions](#) call	10	Developer, Enterprise	Yes
Maximum number of fields/attributes for query suggestions per [GetQuerySuggestions](#) call	10	Developer, Enterprise	Yes
Maximum number of additional fields/attributes for query suggestions per [GetQuerySuggestions](#) call	5	Developer, Enterprise	Yes
Maximum number of block lists per index	1	Developer, Enterprise	No

Description	Default	Edition	Adjustable
Maximum size of a block list text file	2 MB	Developer, Enterprise	Yes
Maximum number of items (words or phrases) in a block list	20,000	Developer, Enterprise	Yes
Maximum number of spell-corrected query suggestions to return in a `Query` API call.	1	Developer, Enterprise	Yes

Document quotas

Description	Default	Edition	Adjustable
Amount of text extracted for an index in a single unit (Developer). You can't add extra units for extracting text for the Developer Edition.	3 GB	Developer	No
Amount of text extracted for an index in a single unit (Enterprise). You can add up to 100 extra units for extracting text for the Enterpris	30 GB	Enterprise	Yes

Description	Default	Edition	Adjustable
e Edition, or simply contact Support.			
Maximum size of a single document or raw file when using a data source connector	50 MB	Developer, Enterprise	Yes
Maximum size of a single document or raw file when using the `BatchPutDocument` API	5 MB	GenAI Enterprise, Developer, Enterprise	Yes
Maximum amount of text extracted from a single document	5 MB	Developer, Enterprise	No
Maximum number of custom fields/attributes per index	500	Developer, Enterprise	No

Featured search results quotas

Description	Default	Edition	Adjustable
Maximum number of featured documents per featured results set	4	Enterprise	Yes
Maximum number of query texts per featured results set	49	Enterprise	No

Description	Default	Edition	Adjustable
Maximum number of characters per query text in a featured results set	1000	Enterprise	Yes
Maximum number of featured results sets per index	50	Enterprise	Yes

Rescore/rerank search results quotas

Description	Default	Edition	Adjustable
Maximum number of `Rescore` requests per second for a rescore execution plan or a single unit of capacity. You can add up to 1000 extra units.	0.01	Enterprise	No
Maximum number of rescore execution plans per account.	50	Enterprise	Yes
Maximum number of tokens in `Title` for a document in a `Rescore` request.	100	Enterprise	No
Maximum number of tokens in Body	200	Enterprise	No

Description	Default	Edition	Adjustable
for a document in a `Rescore` request.			
Maximum number of documents in a `Rescore` request.	25	Enterprise	No
Maximum number of documents per group in a `Rescore` request.	3	Enterprise	No

For more information about Amazon Kendra service quotas and to request a quota increase, see [Service Quotas](#).

Troubleshooting

This section can help you solve common problems you might find when working with Amazon Kendra.

Topics

Troubleshooting data sources

This section can help you solve common issues when configuring and using Amazon Kendra data source connectors.

My documents were not indexed

When you synchronize your Amazon Kendra index with a data source, you may run into issues that prevent the documents from being indexed. Indexing is a two-step process. First, the data source is checked for new and updated documents to index, and to find documents to remove from the index. Second, at the document level, each document is accessed and indexed.

An error can occur in either of these steps. Data source level errors are reported in the console in the **Sync run history** section of the data source details page. The status of the synchronization job can be **Succeeded**, **Incomplete**, or **Failed**. You can also see the number of documents indexed and deleted during the job. If the status is **Failed**, a message is shown in the **Details** column.

Document level errors are reported in Amazon CloudWatch Logs. You can see the errors using the CloudWatch console.

To generate a document sync status report, see I want to generate a sync status report for my documents.

My synchronization job failed

A synchronization job typically fails when there is a configuration error in the index or the data source. In the console, you can find the error message in the **Sync run history** section of the data source details page, under the **Details** column. Document level errors are reported in Amazon CloudWatch Logs. The error message gives information about what went wrong. The problem is usually that the index or the data source does not have the proper IAM permissions. The error message describes the missing permissions. Here are some of the error messages that you can receive:

```
Failed to create log group for job. Please make sure that the IAM role
provided has sufficient permissions.
```

If your index role does not have permission to use CloudWatch, the data source will not be able to create a CloudWatch log. If you get this error, you must add CloudWatch permissions to the index role.

```
Failed to access Amazon S3 file prefix (bucket name) while trying to crawl
your metadata files. Please make sure the IAM role (ARN) provided has
sufficient permissions.
```

When you are using an Amazon S3 data source, Amazon Kendra must have permission to access the bucket that contains the documents. You need to add permission for Amazon Kendra to read the bucket to the data source IAM role.

```
The provided IAM role (ARN) could not be assumed. Please make sure Amazon
Kendra is a trusted entity that is allowed to assume the role.
```

Amazon Kendra needs permission to assume the index and data source IAM roles. You need to add a trust policy to the roles with permission for the `sts:AssumeRole` action.

For the IAM policies that Amazon Kendra needs to index a data source, see IAM roles.

To generate a document sync status report, see I want to generate a sync status report for my documents.

My synchronization job is incomplete

Jobs are generally incomplete when they have completed the data source level process but have some error during the document level process. When a job is incomplete, some of the documents might not have successfully indexed. For an Amazon S3 data source, an incomplete job is typically caused by:

- The metadata for one or more documents was invalid.

- When documents are submitted for indexing but at least one document was not submitted.
- When documents are submitted for deleting from the index but at least one document was not submitted.

To troubleshoot an incomplete synchronization job, look first to your CloudWatch logs.

1. From the details column, choose **View details in CloudWatch**.
2. Review the error messages to see what caused the document to fail.

To generate a document sync status report, see I want to generate a sync status report for my documents.

My synchronization job succeeded but there are no indexed documents

Occasionally, an index synchronization job run will be marked as **Succeeded** but there are no new or updated documents indexed when you expect them. Possible reasons include:

- Check CloudWatch `DocumentsSubmittedForIndexingFailed` metric to see if any documents failed to synchronize. Check your CloudWatch logs for details.
- For an Amazon S3 data source, you may have given Amazon Kendra the wrong bucket name or prefix. Make sure that the bucket that Amazon Kendra is using is the one that contains the documents to index.
- When re-indexing a document that failed to be indexed in an earlier job, Amazon Kendra won't index it unless you've changed the document or its associated metadata file.

To generate a document sync status report, see I want to generate a sync status report for my documents.

I am running into file format issues while syncing my data source

If you run into file format issues while adding files to your data source or syncing your data source, make sure that your document types are Amazon Kendra supported. For a list of document types supported by Amazon Kendra see Document types or formats.

If you are using the `BatchPutDocument` API with plain text files, specify `PLAIN_TEXT` as content type.

I want to generate a sync history report for my documents

You can view a document-level sync run history report in CloudWatch for your data source sync job by selecting **View Report**. A sync run history report will have details about the progress and status of each document in the sync job. It shows if a document succeeded, failed, or was skipped during the crawl, sync, and index stages. You'll also find any error messages related to failed or skipped documents. If the report doesn't show results for an in-progress sync job, the logs may not be available yet. Check back later as data is emitted to the report as events occur during the sync process.

To access your sync run history report, take the following steps:

1. Open the Amazon Kendra console at https://console.aws.amazon.com/kendra/.
2. From the left navigation menu, under **Data management**, choose **Data sources**, and then choose your data source.
3. From your data source summary page, scroll down and select the **Sync history** tab.
4. From **Sync run history**, select **Actions**.
5. From **Actions**, select **View report**. You will be redirected to the CloudWatch console where you will be able to access your report.

> **Note**
>
> A sync run history records if a document was successfully indexed during ingestion, including attached ACLs and metadata, for all Amazon Kendra supported connectors.

If you're using the Amazon S3 connector:

In addition to the vieing the document-level sync run history report in CloudWatch, you can generate sync history reports for each document in your Amazon S3 data source and copy it to an Amazon S3 bucket. During this process, your data is encrypted using AWS KMS keys and can only be viewed by you. Reported document status can be one of the following: **Failed**, **Completed**, or **Succeeded with errors**. Before you can generate sync status reports for Amazon S3, you must do the following:

- Add the following Amazon Kendra service principal to your Amazon S3 access policy

```
{
    “Version”: “2012-10-17",
    “Statement”: [
        {
            “Sid”: “KendraS3Access”,
            “Effect”: “Allow”,
            “Principal”: {
                “Service”: “kendra.amazonaws.com”
            },
            “Action”: “s3:PutObject”,
            “Resource”: “arn:aws:s3:::your-manifest-bucket-name/*”
        }
    ]
}
```

- Create an Amazon S3 bucket with access permissions to Amazon Kendra

If you use the console, to generate a sync history report for Amazon S3, choose to activate the **Generate reports** option from **Sync history reports – *optional*** section on the **Data source details** page. Then, enter the Amazon S3 bucket location and choose from the configuration options available. Reports will be generated from the next sync after you have activated generate report.

If you delete the Amazon S3 bucket, you will lose your log data and will have to set up a new bucket to store new sync reports.

> **Note**
>
> A sync history report provides information only about whether an Amazon S3 connector successfully crawled and ingested data.

How much time does syncing a data source take?

If there are no updates to documents, sync time for a Amazon Kendra index increases in linear proportion to the number of documents. For example, 1,000 documents without any updates would take about five minutes to sync and 2,000 documents without any updates will take about 10 minutes. If there are any updates to the documents, then the sync time will increase based on the number of documents updated.

What is the charge for syncing a data source?

When you sync your index, it takes two minutes to warm up and activate Amazon EC2 to establish the necessary connections. You are not charged during this process. Your usage meter begins only after the sync job starts. For more information on Amazon Kendra pricing, see *Amazon Kendra pricing*.

I am getting an Amazon EC2 authorization error

If an Amazon EC2 unauthorized operation error occurs during a sync for a virtual private cloud (VPC) data source, it's likely that your VPC IAM role lacks required permissions. Please check that the IAM role you use for your data source has the attached permissions. For more information, see Virtual private cloud IAM role.

I am unable to use search index links to open my Amazon S3 objects

Your Amazon Kendra index can only access files that an Amazon S3 data source grants it permissions to access. For example, Amazon Kendra cannot modify the Amazon S3 permissions that determine if an object is meant to be public or encrypted. Amazon Kendra also doesn't have the default permissions to create or return a signed link for Amazon S3 objects. If you want to activate signed linking for Amazon S3 objects in a Amazon Kendra index, you have two options:

- You can use sign your index query results with the source uri object before returning the result to the search page. For a step-by-step walkthrough of this process, see *Sharing objects using presigned URLs*.
- You can override the Amazon S3 object metadata source uri and make your service available through an CloudFront content delivery network (CDN) connected to an Amazon S3 bucket. Or, you can use an API Gateway proxy endpoint that returns a presigned URL and redirect to it.

I am getting an AccessDenied When Using SSL Certificate File error message

If you are getting an access denied error when using an SSL certificate with your data source, make sure that your IAM role has the permission to access the SSL certificate file in its specified location. If the certificate is encrypted with an AWS KMS key, your IAM role should also have permission to decrypt using the AWS KMS key. For more information, see *Authentication and access control for AWS KMS*.

I am getting an authorization error when using a SharePoint data source

If you are getting an authorization error while syncing your index with a SharePoint data source, confirm that you have a Site Admin role assigned to you in SharePoint.

My index does not crawl documents from my Confluence data source

If your Amazon Kendra index is not crawling documents from your Confluence data source during the syncing process, confirm that you are part of Administrator Groups in Confluence.

Troubleshooting document search results

This section can help you fix issues in your Amazon Kendra search results.

My search results are not relevant to my search query

If your search results seem irrelevant, it might be for the following reasons:

- Results with `LOW` confidence are included in the results. You can filter out results with `LOW` confidence by using the QueryResultItem's `ScoreAttributes` field to exclude any result with a value of `LOW`. Amazon Kendra assigns each result a confidence bucket value of either `VERY_HIGH`, `HIGH`, `MEDIUM` and `LOW`. These values indicate the level of confidence that a result is relevant to a query. Also, irrespective of confidence buckets, Amazon Kendra returns three types of results in the following order: `ANSWER` (suggested answer excerpt), `QUESTION_ANSWER` (FAQ) and `DOCUMENT` (document excerpt). Therefore, it is possible for a `LOW` confidence `QUESTION_ANSWER` result to be positioned above a `VERY_HIGH` confidence `DOCUMENT` result. However, it isn't always necessarily true that `LOW` confidence `QUESTION_ANSWER` is a better result than the `VERY_HIGH` confidence `DOCUMENT`.
- Certain metadata fields or attributes are boosted to a very high value, affecting the ranking of results. Amazon Kendra searches your index using multiple parameters such as document title, text, date, and custom text fields or attributes. You can experiment with different boosting values to get the best results across all queries. You can also use dynamic relevance tuning at the query level to use different boosting values for each query.
- Your users are using specialized terms when they query for information and there's no custom synonyms set up for your index to handle these specialized terms. For more details on how and when to use synonyms, see Adding custom synonyms to an index.

Why do I only see 100 results?

Amazon Kendra returns the total count of relevant documents. The top 100 are returned per query by default. The results are paginated. You can use `PageNumber` to access different pages.

You can configure Amazon Kendra to return up to 1,000 documents or search results per query, with up to 100 results per page. To return more than 100 results, you can request this by contacting Quotas Support. Increasing the number of search results could impact latency.

Why are documents that I expect to see missing?

Amazon Kendra supports access control lists (ACLs) based on user and groups. Amazon Kendra ingests ACL policies via connectors. If an index does not configure an ACL, only documents matching the attribute filter for user and group will be shown. If a user or group attribute filter is provided, documents without an ACL will not be shown.

If you are using token-based access control, documents without an ACL policy and documents that match the user and groups will be shown.

Why do I see documents that have an ACL policy?

If an index does not configure an access control policy, then user and groups can be provided by the filter. If no user and group filter is applied, then all related documents will be returned. Any ACL policy will be ignored.

Troubleshooting general issues

Amazon Kendra uses CloudWatch metrics and logs to provide insight into synchronizing your data sources. You can use the metrics and logs to determine what went wrong with a synchronization run and how to fix it.

For general troubleshooting, start with your CloudWatch metrics.

- Check the `DocumentsCrawled` metric to see how many documents your data source checked. For an Amazon S3 bucket, if the number is less than you expect, check that your data source is pointing to the right bucket.
- Check the `DocumentsSkippedNoChange` metric to see how many documents were skipped because they haven't changed since the last synchronization. If the number does not match what you expect, check that your repository was updated correctly.

- Check the `DocumentsSkippedInvalidMetadata` metric to see how many documents had invalid metadata. Check your CloudWatch logs to see the specific errors that occurred.
- Check the `DocumentsSubmittedForIndexingFailed` metric to see how many documents were sent from the data source to the index but failed to be indexed. For example, if you use a metadata attribute in an Amazon S3 data source that hasn't been defined as a custom index field, the document will not be indexed. Check your CloudWatch logs to see the specific errors that occurred.
- Check the `DocumentsSubmittedForDeletionFailed` metric to see how many documents that the data source attempted to remove from the index failed to be deleted from the index. Check your CloudWatch logs to see the specific errors that occurred.

You can look at the CloudWatch logs for a particular synchronization run to get details of the errors that occurred during the run. For more information about CloudWatch logs with Amazon Kendra, see CloudWatch Logs.

Amazon Kendra Intelligent Ranking

Amazon Kendra Intelligent Ranking uses Amazon Kendra semantic search capabilities to intelligently re-rank a search service's results.

Topics

Amazon Kendra Intelligent Ranking for self-managed OpenSearch

You can leverage Amazon Kendra's semantic search capabilities to improve search results from OpenSearch, the self managed open source search service based on the Apache 2.0 License. The Amazon Kendra Intelligent Ranking plugin semantically re-ranks OpenSearch's results using Amazon Kendra. It does this by understanding the meaning and context of a search query using specific fields, such as the document body or title, from the default OpenSearch search results.

Take, for example, this query: "main keynote address". Since 'address' has several meanings, Amazon Kendra can infer the meaning behind the query to return relevant information aligned with the intended meaning. In this context, it's a conference keynote address. A simpler search service might not take into account the intent and could possibly return results for a street address on Main Street, for example.

The Intelligent Ranking plugin for OpenSearch is available for OpenSearch (self managed) version 2.4.0 and later. You can install the plugin using a quick start Bash script to build a new Docker image of OpenSearch with the Intelligent Ranking plugin included. See Setting up the intelligent search plugin—this is an example of a setup to get you up and running quickly.

How the intelligent search plugin works

The overall process of the Intelligent Ranking plugin for OpenSearch (self managed) is as follows:

1. An OpenSearch user issues a query, and OpenSearch provides a query response or a list of documents that are relevant to the query.
2. The Intelligent Ranking plugin takes the query response and extracts information from the documents.

3. The Intelligent Ranking plugin makes a call to Amazon Kendra Intelligent Ranking's Rescore API.
4. The `Rescore` API takes the extracted information from the documents and semantically re-ranks the search results.
5. The `Rescore` API sends the re-ranked search results back to the plugin. The plugin re-arranges the search results in the OpenSearch search response to reflect the new semantic ranking.

The Intelligent Ranking plugin re-ranks results using the "body" and "title" fields. These plugin fields can be mapped to fields in your OpenSearch index that would most fit the definition of a document body and title. For example, if your index contains chapters of a book with fields like "chapter_heading" and "chapter_contents", you can map the former to "title" and the latter to "body" to get the best results.

Setting up the intelligent search plugin

The following outlines how to quickly set up OpenSearch (self managed) with the Intelligent Ranking plugin.

Setting up OpenSearch (self managed) with the Intelligent Ranking plugin (quick setup)

If you are already using Docker image `opensearch:2.4.0`, you can use this Dockerfile to build a new image of OpenSearch 2.4.0 with the Intelligent Ranking plugin. You include a container for the new image in your docker-compose.yml file or opensearch.yml file. You also include your generated rescore execution plan ID from creating a rescore execution plan, along with your region and endpoint information—see step 2 for creating a rescore execution plan.

If you had previously downloaded a version of the `opensearch` Docker image that's older than 2.4.0, you must use Docker image `opensearch:2.4.0` or later and build a new image with the Intelligent Ranking plugin included.

1. Download and install Docker Desktop for your operating system. Docker Desktop includes Docker Compose and Docker Engine. It's recommended that you check whether your computer meets the system requirements mentioned in the Docker installation details.

   You can also increase your memory usage requirements within the settings of your Docker Desktop. You are responsible for the usage requirements of Docker outside the freely available usage limits for Docker services. See Docker subscriptions.

   Check Docker Desktop status is "running".

2. Provision Amazon Kendra Intelligent Ranking and your capacity requirements. Once you provision Amazon Kendra Intelligent Ranking, you are charged hourly based on your set capacity units. See free tier and pricing information.

   You use the CreateRescoreExecutionPlan API to provision the `Rescore API`. If you don't need more capacity units than the single unit default, don't add more units and provide only a name for your rescore execution plan. You can also update your capacity requirements by using the UpdateRescoreExecutionPlan API. For more information, see Semantically ranking a search service's results.

   Optionally, you can go to step 3 to create a default rescore execution plan when you run the quick start Bash script.

   Note for step 4 the rescore execution plan ID included in the response.

   CLI

```
aws kendra-ranking create-rescore-execution-plan \
  --name MyRescoreExecutionPlan \
  --capacity-units '{"RescoreCapacityUnits":<integer number of additional
 capacity units>}'

Response:

{
    "Id": "<rescore execution plan ID>",
    "Arn": "arn:aws:kendra-ranking:<region>:<account-id>:rescore-execution-plan/
<rescore-execution-plan-id>"
}
```

   Python

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra_ranking = boto3.client("kendra-ranking")

print("Create a rescore execution plan.")
```

```
# Provide a name for the rescore execution plan
name = "MyRescoreExecutionPlan"
# Set your required additional capacity units
# Don't set capacity units if you don't require more than 1 unit given by
 default
capacity_units = 1

try:
    rescore_execution_plan_response =
 kendra_ranking.create_rescore_execution_plan(
        Name = name,
        CapacityUnits = {"RescoreCapacityUnits":capacity_units}
    )

    pprint.pprint(rescore_execution_plan_response)

    rescore_execution_plan_id = rescore_execution_plan_response["Id"]

    print("Wait for Amazon Kendra to create the rescore execution plan.")

    while True:
        # Get the details of the rescore execution plan, such as the status
        rescore_execution_plan_description =
 kendra_ranking.describe_rescore_execution_plan(
            Id = rescore_execution_plan_id
        )
        # When status is not CREATING quit.
        status = rescore_execution_plan_description["Status"]
        print(" Creating rescore execution plan. Status: "+status)
        time.sleep(60)
        if status != "CREATING":
            break

except ClientError as e:
        print("%s" % e)

print("Program ends.")
```

3. Download the quick start Bash script from GitHub for your version of OpenSearch by selecting the version branch from the main branch dropdown.

   This script uses Docker images for OpenSearch and OpenSearch Dashboards using your version you selected on the GitHub repository for the script. It downloads a zip file for the Intelligent

Ranking plugin, and generates a `Dockerfile` to build a new Docker image of OpenSearch that includes the plugin. It also creates a [docker-compose.yml](#) file that includes containers for OpenSearch with the Intelligent Ranking plugin and OpenSearch Dashboards. The script adds your rescore execution plan ID, region information, and endpoint (uses the region) to the docker-compose.yml file. The script then runs `docker-compose up` to start the containers for OpenSearch with Intelligent Ranking included and OpenSearch Dashboards. To stop the containers without removing them, run `docker-compose stop`. To remove the containers, run `docker-compose down`.

4. Open your terminal and in the directory of the Bash script, run the following command.

```
bash search_processing_kendra_quickstart.sh -p <execution-plan-id> -r <region>
```

When you run this command, you provide the rescore execution plan ID that you noted in step 2 when you provisioned Amazon Kendra Intelligent Ranking, along with your region information. Optionally, you can instead provision Amazon Kendra Intelligent Ranking by using the `--create-execution-plan` option. This creates a rescore execution plan with a default name and default capacity.

To not lose your index when the default ephemeral container is removed, you can have your index persist across executions by providing the data volume name using the `--volume-name` option. If you previously created an index, you can specify the volume in your docker-compose.yml or opensearch.yml file. To leave your volumes intact, **don't** run `docker-compose down -v`.

The quick start Bash script configures your AWS credentials in the OpenSearch keystore to connect to Amazon Kendra Intelligent Ranking. To provide your AWS credentials to the script, use the `--profile` option to specify the AWS profile. If the `--profile` option is not specified, then the quick start Bash script attempts to read AWS credentials (access/secret key, optional session token) from environment variables, and then from the default AWS profile. If the `--profile` option is not specified and no credentials are found, the script won't pass credentials to the OpenSearch keystore. If no credentials are specified in the OpenSearch keystore, then the plugin still checks credentials in the [Default Credential Provider Chain](#), including Amazon ECS container credentials or instance profile credentials delivered through the Amazon EC2 metadata service.

Make sure that you have created an IAM role with the necessary permissions to invoke Amazon Kendra Intelligent Ranking. The following is an example of an IAM policy to grant permission to use the `Rescore` API for a specific rescore execution plan:

Example of docker-compose.yml

An example of a docker-compose.yml file using OpenSearch 2.4.0 or later with the Intelligent Ranking plugin and OpenSearch Dashboards 2.4.0 or later.

```
version: '3'
networks:
  opensearch-net:
volumes:
  <volume-name>:
services:
  opensearch-node:
    image: <Docker image tag name of OpenSearch with Intelligent Ranking plugin>
    container_name: opensearch-node
    environment:
      - cluster.name=opensearch-cluster
      - node.name=opensearch-node
      - discovery.type=single-node
      - kendra_intelligent_ranking.service.endpoint=https://kendra-
ranking.<region>.api.aws
      - kendra_intelligent_ranking.service.region=<region>
      - kendra_intelligent_ranking.service.execution_plan_id=<rescore-execution-plan-
id>
    ulimits:
      memlock:
        soft: -1
        hard: -1
      nofile:
        soft: 65536
        hard: 65536
    ports:
      - 9200:9200
      - 9600:9600
    networks:
      - opensearch-net
  volumes:
    <docker-volume-name>:/usr/share/opensearch/data
```

```
  opensearch-dashboard:
    image: opensearchproject/opensearch-dashboards:<your-version>
    container_name: opensearch-dashboards
    ports:
      - 5601:5601
    environment:
      OPENSEARCH_HOSTS: '["https://opensearch-node:9200"]'
    networks:
      - opensearch-net
```

Example of a Dockerfile and building an image

An example of a `Dockerfile` for using OpenSearch 2.4.0 or later with the Intelligent Ranking plugin.

```
FROM opensearchproject/opensearch:<your-version>
RUN /usr/share/opensearch/bin/opensearch-plugin install --batch  https://github.com/
opensearch-project/search-processor/releases/download/<your-version>/search-
processor.zip
```

Building a Docker image for OpenSearch with the Intelligent Ranking plugin.

```
docker build --tag=<Docker image tag name of OpenSearch with Intelligent Ranking
 plugin>
```

Interacting with the intelligent search plugin

Once you have set up OpenSearch (self managed) with the Intelligent Ranking plugin, you can interact with the plugin using curl commands or OpenSearch client libraries. The default credentials for accessing OpenSearch with the Intelligent Ranking plugin are user name 'admin' and password 'admin'.

To apply the Intelligent Ranking plugin settings to an OpenSearch index:

Curl

```
curl -XPUT "https://localhost:9200/<your-docs-index>/_settings" -u 'admin:admin' --
insecure -H 'Content-Type: application/json' -d'
{
  "index": {
    "plugin" : {
```

```
      "searchrelevance" : {
        "result_transformer" : {
          "kendra_intelligent_ranking": {
              "order": 1,
              "properties": {
                "title_field": "title_field_name_here",
                "body_field": "body_field_name_here"
              }
          }
        }
      }
    }
  }
}
'
```

Python

```
pip install opensearch-py

from opensearchpy import OpenSearch
host = 'localhost'
port = 9200
auth = ('admin', 'admin')

client = OpenSearch(
    hosts = [{'host': host, 'port': port}],
    http_compress = True, # enables gzip compression for request bodies
    http_auth = auth,
    # client_cert = client_cert_path,
    # client_key = client_key_path,
    use_ssl = True,
    verify_certs = False,
    ssl_assert_hostname = False,
    ssl_show_warn = False,
    ca_certs = ca_certs_path
)

setting_body = {
    "index": {
        "plugin" : {
            "searchrelevance" : {
                "result_transformer" : {
```

```
                    "kendra_intelligent_ranking": {
                            "order": 1,
                            "properties": {
                                "title_field": "title_field_name_here",
                                "body_field": "body_field_name_here"
                            }
                    }
                }
            }
        }
    }
}

response = client.indices.put_settings(index_name, body=setting_body)
```

You must include the name of the main text field you want to use to re-rank on, such as a document body or document contents field. You can also include other text fields, such as document title or document summary.

Now you can issue any query and the results are ranked using the Intelligent Ranking plugin.

Curl

```
curl -XGET "https://localhost:9200/<your-docs-index>/_search?pretty" -u
 'admin:admin' --insecure -H 'Content-Type: application/json' -d'
{
  "query" : {
    "match" : {
      "body_field_name_here": "intelligent systems"
    }
  }
}
'
```

Python

```
from opensearchpy import OpenSearch
host = 'localhost'
port = 9200
auth = ('admin', 'admin')

client = OpenSearch(
```

```
    hosts = [{'host': host, 'port': port}],
    http_compress = True, # enables gzip compression for request bodies
    http_auth = auth,
    # client_cert = client_cert_path,
    # client_key = client_key_path,
    use_ssl = True,
    verify_certs = False,
    ssl_assert_hostname = False,
    ssl_show_warn = False,
    ca_certs = ca_certs_path
)

query = {
  'size': 10,
  "query" : {
    "match" : {
      "body_field_name_here": "intelligent systems"
    }
  }
}

response = client.search(
    body = query,
    index = index_name
)

print('\nSearch results:')
print(response)
```

To remove the Intelligent Ranking plugin settings for an OpenSearch index:

Curl

```
curl -XPUT "http://localhost:9200/<your-docs-index>/_settings" -H 'Content-Type:
 application/json' -d'
{
  "index": {
    "plugin": {
      "searchrelevance": {
        "result_transformer": {
          "kendra_intelligent_ranking.*": null
        }
```

```
      }
    }
  }
}
'
```

Python

```
from opensearchpy import OpenSearch
host = 'localhost'
port = 9200
auth = ('admin', 'admin')

client = OpenSearch(
    hosts = [{'host': host, 'port': port}],
    http_compress = True, # enables gzip compression for request bodies
    http_auth = auth,
    # client_cert = client_cert_path,
    # client_key = client_key_path,
    use_ssl = True,
    verify_certs = False,
    ssl_assert_hostname = False,
    ssl_show_warn = False,
    ca_certs = ca_certs_path
)

setting_body = {
  "index": {
    "plugin": {
      "searchrelevance": {
        "result_transformer": {
          "kendra_intelligent_ranking.*": null
        }
      }
    }
  }
}

response = client.indices.put_settings(index_name, body=setting_body)
```

To test the Intelligent Ranking plugin on a certain query or to test on certain body and title fields:

Curl

```
curl -XGET "https://localhost:9200/<your-docs-index>/_search?pretty" -u
 'admin:admin' --insecure -H 'Content-Type: application/json' -d'
{
  "query": {
    "multi-match": {
      "query": "intelligent systems",
      "fields": ["body_field_name_here", "title_field_name_here"]
    }
  },
  "size": 25,
  "ext": {
    "search_configuration": {
      "result_transformer": {
        "kendra_intelligent_ranking": {
          "order": 1,
          "properties": {
            "title_field": "title_field_name_here",
            "body_field": "body_field_name_here"
          }
        }
      }
    }
  }
}
'
```

Python

```
from opensearchpy import OpenSearch
host = 'localhost'
port = 9200
auth = ('admin', 'admin')

client = OpenSearch(
    hosts = [{'host': host, 'port': port}],
    http_compress = True, # enables gzip compression for request bodies
    http_auth = auth,
    # client_cert = client_cert_path,
    # client_key = client_key_path,
    use_ssl = True,
    verify_certs = False,
```

```
    ssl_assert_hostname = False,
    ssl_show_warn = False,
    ca_certs = ca_certs_path
)

# Index settings null for kendra_intelligent_ranking

query = {
  "query": {
    "multi_match": {
      "query": "intelligent systems",
      "fields": ["body_field_name_here", "title_field_name_here"]
    }
  },
  "size": 25,
  "ext": {
    "search_configuration": {
      "result_transformer": {
        "kendra_intelligent_ranking": {
          "order": 1,
          "properties": {
            "title_field": "title_field_name_here",
            "body_field": "body_field_name_here"
          }
        }
      }
    }
  }
}

response = client.search(
    body = query,
    index = index_name
)

print('\nSearch results:')
print(response)
```

Comparing OpenSearch results with Amazon Kendra results

You can compare side-by-side OpenSearch (self managed) ranked results against Amazon Kendra's re-ranked results. OpenSearch Dashboards version 2.4.0 and later offers side-by-side results so that you can compare how OpenSearch ranks documents with how Amazon Kendra or the plugin ranks documents for a search query.

Before you can compare OpenSearch ranked results against Amazon Kendra re-ranked results, make sure your OpenSearch Dashboards is backed by an OpenSearch server with the Intelligent Ranking plugin. You can set this up using Docker and a quick start Bash script. See Setting up the intelligent search plugin.

The following outlines how to compare OpenSearch and Amazon Kendra search results in OpenSearch Dashboards. For more information, see the OpenSearch Documentation.

Comparing search results in OpenSearch Dashboards

1. Open http://localhost:5601 and sign in to OpenSearch Dashboards. The default credentials are user name 'admin' and password 'admin'.
2. Select **Search Relevance** from the OpenSearch plugins in the navigation menu.
3. Enter the search text in the search bar.
4. Select your index for **Query 1** and enter a query in the OpenSearch Query DSL. You can use the `%SearchText%` variable to refer to the search text you entered in the search bar. For an example of this query, see OpenSearch Documentation. The results returned for this query are the OpenSearch results without using the Intelligent Ranking plugin.
5. Select the same index for **Query 2** and enter the same query in the OpenSearch Query DSL. In addition, include the extension with `kendra_intelligent_ranking` and specify the mandatory `body_field` to rank on. You can also specify the title field, but the body field is mandatory. For an example of this query, see OpenSearch Documentation. The results returned for this query are the Amazon Kendra re-ranked results using the Intelligent Ranking plugin. The plugin ranks up to 25 results.
6. Select **Search** to return and compare results.

Semantically ranking a search service's results

Amazon Kendra Intelligent Ranking uses Amazon Kendra's semantic search capabilities to re-rank a search service's results. It does this by taking into account the search query's context, plus all the available information from the search service documents. Amazon Kendra Intelligent Ranking can improve simple keyword matching.

The CreateRescoreExecutionPlan API creates an Amazon Kendra Intelligent Ranking resource used for provisioning the Rescore API. The `Rescore` API re-ranks search results from a search service such as OpenSearch (self managed).

When you call `CreateRescoreExecutionPlan`, you set your required capacity units for re-ranking a search service's results. If you don't need more capacity units beyond the single unit default, don't change the default. Provide only a name for your rescore execution plan. You can set up to 1000 extra units. For information on what is included in a single capacity unit, see Adjusting capacity. Once you provision Amazon Kendra Intelligent Ranking, you are charged hourly based on your set capacity units. See free tier and pricing information.

A rescore execution plan ID is generated and returned in the response when you call `CreateRescoreExecutionPlan`. The `Rescore` API uses the rescore execution plan ID to re-rank a search service's results using the capacity you set. You include the rescore execution plan ID in the configuration files of your search service. For example, if you use OpenSearch (self managed), you include the rescore execution plan ID in your docker-compose.yml or opensearch.yml file—see Intelligently ranking OpenSearch (self service) results.

An Amazon Resource Name (ARN) is also generated in the response when you call `CreateRescoreExecutionPlan`. You can use this ARN to create a permissions policy in AWS Identity and Access Management (IAM) to restrict user access to a specific ARN for a specific rescore execution plan. For an example of an IAM policy to grant permission to use the `Rescore` API for a specific rescore execution plan, see Amazon Kendra Intelligent Ranking for self-managed OpenSearch.

The following is an example of creating a rescore execution plan with capacity units set to 1.

CLI

```
aws kendra-ranking create-rescore-execution-plan \
  --name MyRescoreExecutionPlan \
  --capacity-units '{"RescoreCapacityUnits":1}'

Response:

{
    "Id": "<rescore execution plan ID>",
    "Arn": "arn:aws:kendra-ranking:<region>:<account-id>:rescore-execution-plan/
<rescore-execution-plan-id>"
}
```

Python

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra_ranking = boto3.client("kendra-ranking")

print("Create a rescore execution plan.")

# Provide a name for the rescore execution plan
name = "MyRescoreExecutionPlan"
# Set your required additional capacity units
# Don't set capacity units if you don't require more than 1 unit given by default
capacity_units = 1

try:
    rescore_execution_plan_response = kendra_ranking.create_rescore_execution_plan(
        Name = name,
        CapacityUnits = {"RescoreCapacityUnits":capacity_units}
    )

    pprint.pprint(rescore_execution_plan_response)

    rescore_execution_plan_id = rescore_execution_plan_response["Id"]

    print("Wait for Amazon Kendra to create the rescore execution plan.")

    while True:
        # Get the details of the rescore execution plan, such as the status
        rescore_execution_plan_description = 
 kendra_ranking.describe_rescore_execution_plan(
            Id = rescore_execution_plan_id
        )
        # When status is not CREATING quit.
        status = rescore_execution_plan_description["Status"]
        print(" Creating rescore execution plan. Status: "+status)
        time.sleep(60)
        if status != "CREATING":
            break

except ClientError as e:
        print("%s" % e)
```

```
print("Program ends.")
```

Java

```
import java.util.concurrent.TimeUnit;

import software.amazon.awssdk.services.kendraranking.KendraRankingClient;
import
 software.amazon.awssdk.services.kendraranking.model.CapacityUnitsConfiguration;
import
 software.amazon.awssdk.services.kendraranking.model.CreateRescoreExecutionPlanRequest;
import
 software.amazon.awssdk.services.kendraranking.model.CreateRescoreExecutionPlanResponse;
import
 software.amazon.awssdk.services.kendraranking.model.DescribeRescoreExecutionPlanRequest;
import
 software.amazon.awssdk.services.kendraranking.model.DescribeRescoreExecutionPlanResponse;
import
 software.amazon.awssdk.services.kendraranking.model.RescoreExecutionPlanStatus;

public class CreateRescoreExecutionPlanExample {

  public static void main(String[] args) throws InterruptedException {

    String rescoreExecutionPlanName = "MyRescoreExecutionPlan";
    int capacityUnits = 1;

    KendraRankingClient kendraRankingClient = KendraRankingClient.builder().build();

    System.out.println(String.format("Creating a rescore execution plan named %s",
 rescoreExecutionPlanName));

    CreateRescoreExecutionPlanResponse createResponse =
 kendraRankingClient.createRescoreExecutionPlan(
        CreateRescoreExecutionPlanRequest.builder()
            .name(rescoreExecutionPlanName)
            .capacityUnits(
                CapacityUnitsConfiguration.builder()
                    .rescoreCapacityUnits(capacityUnits)
                    .build()
            )
            .build()
```

```
    );

    String rescoreExecutionPlanId = createResponse.id();
    System.out.println(String.format("Waiting for rescore execution plan with id %s
 to finish creating.", rescoreExecutionPlanId));
    while (true) {
      DescribeRescoreExecutionPlanResponse describeResponse =
 kendraRankingClient.describeRescoreExecutionPlan(
          DescribeRescoreExecutionPlanRequest.builder()
              .id(rescoreExecutionPlanId)
              .build()
      );
      RescoreExecutionPlanStatus rescoreExecutionPlanStatus =
 describeResponse.status();
      if (rescoreExecutionPlanStatus != RescoreExecutionPlanStatus.CREATING) {
        break;
      }
      TimeUnit.SECONDS.sleep(60);
    }

    System.out.println("Rescore execution plan creation is complete.");
  }
}
```

The following is an example of updating a rescore execution plan to set capacity units to 2.

CLI

```
aws kendra-ranking update-rescore-execution-plan \
  --id <rescore execution plan ID> \
  --capacity-units '{"RescoreCapacityUnits":2}'
```

Python

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra_ranking = boto3.client("kendra-ranking")

print("Update a rescore execution plan.")
```

```
# Provide the ID of the rescore execution plan
id = <rescore execution plan ID>
# Re-set your required additional capacity units
capacity_units = 2

try:
    kendra_ranking.update_rescore_execution_plan(
        Id = id,
        CapacityUnits = {"RescoreCapacityUnits":capacity_units}
    )

    print("Wait for Amazon Kendra to update the rescore execution plan.")

    while True:
        # Get the details of the rescore execution plan, such as the status
        rescore_execution_plan_description =
 kendra_ranking.describe_rescore_execution_plan(
            Id = id
        )
        # When status is not UPDATING quit.
        status = rescore_execution_plan_description["Status"]
        print(" Updating rescore execution plan. Status: "+status)
        time.sleep(60)
        if status != "UPDATING":
            break

except ClientError as e:
        print("%s" % e)

print("Program ends.")
```

Java

```
import java.util.concurrent.TimeUnit;

import software.amazon.awssdk.services.kendraranking.KendraRankingClient;
import
 software.amazon.awssdk.services.kendraranking.model.CapacityUnitsConfiguration;
import
 software.amazon.awssdk.services.kendraranking.model.DescribeRescoreExecutionPlanRequest;
import
 software.amazon.awssdk.services.kendraranking.model.DescribeRescoreExecutionPlanResponse;
```

```
import
 software.amazon.awssdk.services.kendraranking.model.RescoreExecutionPlanStatus;
import
 software.amazon.awssdk.services.kendraranking.model.UpdateRescoreExecutionPlanRequest;
import
 software.amazon.awssdk.services.kendraranking.model.UpdateRescoreExecutionPlanResponse;

public class UpdateRescoreExecutionPlanExample {

  public static void main(String[] args) throws InterruptedException {

    String rescoreExecutionPlanId = <rescore execution plan ID>;
    int newCapacityUnits = 2;

    KendraRankingClient kendraRankingClient = KendraRankingClient.builder().build();

    System.out.println(String.format("Updating a rescore execution plan named %s",
 rescoreExecutionPlanId));

    UpdateRescoreExecutionPlanResponse updateResponse =
 kendraRankingClient.updateRescoreExecutionPlan(
        UpdateRescoreExecutionPlanRequest.builder()
            .id(rescoreExecutionPlanId)
            .capacityUnits(
                CapacityUnitsConfiguration.builder()
                    .rescoreCapacityUnits(newCapacityUnits)
                    .build()
            )
            .build()
    );

    System.out.println(String.format("Waiting for rescore execution plan with id %s
 to finish updating.", rescoreExecutionPlanId));
    while (true) {
      DescribeRescoreExecutionPlanResponse describeResponse =
 kendraRankingClient.describeRescoreExecutionPlan(
          DescribeRescoreExecutionPlanRequest.builder()
              .id(rescoreExecutionPlanId)
              .build()
      );
      RescoreExecutionPlanStatus rescoreExecutionPlanStatus =
 describeResponse.status();
      if (rescoreExecutionPlanStatus != RescoreExecutionPlanStatus.UPDATING) {
        break;
```

```
        }
        TimeUnit.SECONDS.sleep(60);
      }

      System.out.println("Rescore execution plan update is complete.");
    }
}
```

The following is an example of using the `Rescore` API.

CLI

```
aws kendra-ranking rescore \
  --rescore-execution-plan-id <rescore execution plan ID> \
  --search-query "intelligent systems" \
  --documents "[{\"Id\": \"DocId1\",\"Title\": \"Smart systems\", \"Body\":
 \"intelligent systems in everyday life\",\"OriginalScore\": 2.0}, {\"Id\":
 \"DocId2\",\"Title\": \"Smarter systems\", \"Body\": \"living with intelligent
 systems\",\"OriginalScore\": 1.0}]"
```

Python

```
import boto3
from botocore.exceptions import ClientError
import pprint

kendra_ranking = boto3.client("kendra-ranking")

print("Use the Rescore API.")

# Provide the ID of the rescore execution plan
id = <rescore execution plan ID>
# The search query from the search service
query = "intelligent systems"
# The list of documents for Intelligent Ranking to rescore
document_list = [
    {"Id": "DocId1", "Title": "Smart systems", "Body": "intelligent systems in
 everyday life", "OriginalScore": 2.0},
    {"Id": "DocId2", "Title": "Smarter systems", "Body": "living with intelligent
 systems", "OriginalScore": 1.0}
]
```

```
try:
    rescore_response = kendra_ranking.rescore(
        rescore_execution_plan_id = id,
        search_query = query,
        documents = document_list
    )

    print(rescore_response["RescoreId"])
    print(rescore_resposne["ResultItems"])

except ClientError as e:
        print("%s" % e)

print("Program ends.")
```

Java

```
import java.util.ArrayList;
import java.util.List;

import software.amazon.awssdk.services.kendraranking.KendraRankingClient;
import software.amazon.awssdk.services.kendraranking.model.RescoreRequest;
import software.amazon.awssdk.services.kendraranking.model.RescoreResponse;
import software.amazon.awssdk.services.kendraranking.model.Document;

public class RescoreExample {

  public static void main(String[] args) {

    String rescoreExecutionPlanId = <rescore execution plan ID>;
    String query = "intelligent systems";

    List<Document> documentList = new ArrayList<>();
    documentList.add(
        Document.builder()
            .id("DocId1")
            .originalScore(2.0F)
            .body("intelligent systems in everyday life")
            .title("Smart systems")
            .build()
    );
    documentList.add(
        Document.builder()
```

```
                .id("DocId2")
                .originalScore(1.0F)
                .body("living with intelligent systems")
                .title("Smarter systems")
                .build()
        );

        KendraRankingClient kendraRankingClient = KendraRankingClient.builder().build();

        RescoreResponse rescoreResponse = kendraRankingClient.rescore(
            RescoreRequest.builder()
                .rescoreExecutionPlanId(rescoreExecutionPlanId)
                .searchQuery(query)
                .documents(documentList)
                .build()
        );

        System.out.println(rescoreResponse.rescoreId());
        System.out.println(rescoreResponse.resultItems());
    }
}
```

API reference

The API reference documentation is now a separate guide.

Document history for Amazon Kendra

- **Latest documentation update:** December 4, 2024

The following table describes important changes in each release of Amazon Kendra. For notification about updates to this documentation, you can subscribe to the RSS feed.

Change	Description	Date
New feature	Amazon Kendra now supports a new index type: the Amazon Kendra GenAI Enterprise Edition index. For more information, see Amazon Kendra GenAI Enterprise Edition index.	December 4, 2024
New feature	Amazon Kendra now supports an updated version of the GitHub data source connector . For more information, see GitHub.	February 27, 2024
New feature	Amazon Kendra now supports an updated versions of the Amazon FSx data source connector. For more information, see Amazon FSx (Windows) and Amazon FSx (NetApp ONTAP).	February 8, 2024
New feature	Amazon Kendra now supports an updated version of the Slack data source connector . For more information, see Slack.	January 11, 2024

New feature	Amazon Kendra now supports collapsing and expanding your search results. For more information, see Collapsing/expanding search results.	October 19, 2023
New feature	Amazon Kendra now supports an Aurora (MySQL) data source connector. For more information, see Aurora (MySQL).	September 28, 2023
New feature	Amazon Kendra now supports an Aurora (PostgreSQL) data source connector. For more information, see Aurora (PostgreSQL).	September 28, 2023
New feature	Amazon Kendra now supports an Amazon RDS (MySQL) data source connector. For more information, see Amazon RDS (MySQL).	September 28, 2023
New feature	Amazon Kendra now supports an Amazon RDS (Microsoft SQL Server) data source connector. For more information, see Amazon RDS (Microsoft SQL Server).	September 28, 2023
New feature	Amazon Kendra now supports an Amazon RDS (Oracle) data source connector. For more information, see Amazon RDS (Oracle).	September 28, 2023

New feature	Amazon Kendra now supports an Amazon RDS (PostgreS QL) data source connector . For more information, see Amazon RDS (PostgreSQL).	September 28, 2023
New feature	Amazon Kendra now supports a IBM DB2 data source connector. For more informati on, see IBM DB2.	September 28, 2023
New feature	Amazon Kendra now supports a Microsoft SQL Server data source connector. For more information, see Microsoft SQL Server.	September 28, 2023
New feature	Amazon Kendra now supports a MySQL data source connector. For more informati on, see MySQL.	September 28, 2023
New feature	Amazon Kendra now supports an Oracle Database data source connector. For more information, see Oracle Database.	September 28, 2023
New feature	Amazon Kendra now supports a PostgreSQL data source connector. For more informati on, see PostgreSQL.	September 28, 2023
New feature	Amazon Kendra now provides a data source connector for Drupal. For more information, see Drupal.	September 6, 2023

New feature	Retrieve semantically relevant passages using the Amazon Kendra Retrieve API for retrieval augmented generation (RAG) systems.	June 22, 2023
New feature	Amazon Kendra now supports an updated version of the Amazon Kendra Web Crawler data source connector. For more information, see Amazon Kendra Web Crawler v2.0.	June 21, 2023
Region expansion	Amazon Kendra is now available in Europe (London) (eu-west-2).	June 5, 2023
New feature	Amazon Kendra now supports an updated version of the Alfresco data source connector. For more informati on, see Alfresco.	May 16, 2023
New feature	Amazon Kendra now provides a data source connector for Adobe Experience Manager. For more information, see Adobe Experience Manager.	May 11, 2023

New feature	Amazon Kendra now supports configuring document fields/attributes when you call GetQuerySuggestions. You can now base query suggestions on the contents of document fields. For more information, see Query suggestions.	May 2, 2023
New feature	Amazon Kendra now provides a data source connector for Gmail. For more information, see Gmail.	April 13, 2023
New feature	Amazon Kendra now supports an updated version of the Microsoft OneDrive data source connector. For more information, see Microsoft OneDrive v2.0.	April 3, 2023
New feature	Improve the visibility of new documents or promote certain documents when your users type certain queries using Featured results.	March 30, 2023
New feature	Amazon Kendra now supports an updated data source connector for Microsoft SharePoint. For more information, see Microsoft SharePoint.	March 2, 2023

New feature	Amazon Kendra now supports an updated version of the Confluence data source connector. For more informati on, see Confluence.	March 1, 2023
Region expansion	Amazon Kendra is now available in Asia Pacific (Tokyo) (ap-northeast-1).	February 7, 2023
New feature	Amazon Kendra now provides a data source connector for Microsoft Exchange. For more information, see Microsoft Exchange.	January 12, 2023
New feature	Amazon Kendra now provides a data source connector for Microsoft Yammer. For more information, see Microsoft Yammer.	January 12, 2023
New feature	Amazon Kendra now supports indexing RTF, XML, XSLT, MS_EXCEL, CSV, JSON, and MD document types. For more information, see Types of documents.	January 11, 2023
New feature	Amazon Kendra now supports an updated version of the Amazon S3 data source connector. For more informati on, see Amazon S3.	January 10, 2023

New feature	OpenSearch (self managed) search results can be semantically ranked using Amazon Kendra Intelligent Ranking.	January 9, 2023
New feature	Amazon Kendra now provides a data source connector for Microsoft Teams. For more information, see Microsoft Teams.	January 5, 2023
New feature	Amazon Kendra has an updated data source connector for Google Drive. For more information, see Google Drive .	January 5, 2023
New feature	Amazon Kendra has an updated data source connector for ServiceNow. For more information, see ServiceNow.	December 21, 2022
New feature	Amazon Kendra has an updated data source connector for Salesforce. For more information, see Salesforce.	December 21, 2022
Region expansion	Amazon Kendra is now available in Asia Pacific (Mumbai) (ap-south-1).	December 14, 2022

New feature	Amazon Kendra's tabular search feature can search and extract answers from tables embedded in HTML documents.	November 27, 2022
New feature	Amazon Kendra supports semantic search for a select set of languages.	November 27, 2022
New feature	Amazon Kendra now provides a data source connector for Dropbox. For more informati on, see Dropbox.	September 27, 2022
New feature	Amazon Kendra now provides a data source connector for Zendesk. For more informati on, see Zendesk.	August 17, 2022
New feature	Document level access control can now be re-configured after you index your documents. For more information, see Access control configuration.	July 14, 2022
New feature	Amazon Kendra now provides a data source connector for Alfresco. For more informati on, see Alfresco.	June 30, 2022
New feature	Amazon Kendra now provides a data source connector for GitHub. For more information, see GitHub.	June 2, 2022

New feature	Amazon Kendra now provides a data source connector for Jira. For more information, see Jira.	May 12, 2022
New feature	Nested facets within a facet can be displayed in the search results. For more information, see Facets.	May 5, 2022
New feature	Amazon Kendra now provides a data source connector for Quip. For more information, see Quip.	April 19, 2022
New feature	Amazon Kendra now provides a data source connector for Box. For more information, see Box.	April 6, 2022
New feature	Amazon Kendra now provides a data source connector for Slack. For more information, see Slack.	March 14, 2022
New feature	Amazon Kendra now provides a data source connector for Amazon FSx. For more information, see Amazon FSx.	February 8, 2022
AWS managed policy updates - New policies	Amazon Kendra added new AWS managed policies. For more information, see AWS Managed policies for Amazon Kendra.	January 3, 2022

New feature	Amazon Kendra search application can be deployed in a few clicks without the need for any front-end code. For more information, see Deploying a search application with no code.	December 1, 2021
New feature	Document metadata and content can be enriched during the document ingestion process. For more information, see Customizing document metadata during the ingestion process.	December 1, 2021
New feature	Amazon Kendra offers search analytics to gain useful insights into your search application. For more information, see Gaining insights with search analytics.	December 1, 2021
Region expansion	Amazon Kendra is now available in AWS GovCloud (US-West) (us-gov-west-1).	October 13, 2021
New feature	Amazon Kendra can now index documents in multiple languages and filter search results by language. See Adding documents in languages other than English and Searching in languages.	October 7, 2021

New feature	Amazon Kendra now integrates with Identity Center directory to fetch access levels of groups and users for user context filtering. See User-group configuration for IAM Identity Center.	October 6, 2021
New tutorial	Amazon Kendra now provides a tutorial that walks you through how to build a metadata-enriched search solution. See Building an intelligent search solution.	August 13, 2021
New feature	Amazon Kendra now provides a data source connector for WorkDocs. For more information, see WorkDocs.	July 20, 2021
New feature	Amazon Kendra now provides a web crawler to crawl and index webpages. For more information, see Web crawler.	June 17, 2021
Region expansion	Amazon Kendra is now available in Canada (Central) (ca-central-1).	June 16, 2021
Region expansion	Amazon Kendra is now available in US East (Ohio) (us-east-2).	June 7, 2021

New feature	Amazon Kendra now supports query suggestions, where users are suggested popular queries relevant to their search. For more informati on, see Suggesting popular search queries.	May 27, 2021
AWS managed policy updates - New policies	Amazon Kendra added new AWS managed policies. For more information, see AWS Managed policies for Amazon Kendra.	May 27, 2021
Region expansion	Amazon Kendra is now available in Asia Pacific (Singapore) (ap-southeast-1).	May 5, 2021
New feature	Amazon Kendra now supports tuning search relevance in the query by overriding tuning configurations set at the index level. For more information, see Tuning search relevance and Tuning responses.	April 20, 2021
New feature	Amazon Kendra now supports OAuth 2.0 authentication and using ServiceNow queries to select documents for indexing. For more informati on, see ServiceNow.	April 1, 2021

New feature	Amazon Kendra now supports incremental learning for FAQ documents. For more information, see Submitting feedback for incremental learning.	February 17, 2021
New feature	Amazon Kendra now supports index synonyms. For more information, see Adding synonyms to an index.	December 10, 2020
New feature	Amazon Kendra now provides a data base connector for Google Workspace Drive. For more information, see Using a Google Workspace Drive data source.	December 8, 2020
New feature	Amazon Kendra now provides a JavaScript library that makes it easier for you to provide query feedback to Amazon Kendra. For more information, see Submitting feedback.	December 8, 2020
New feature	Amazon Kendra now supports token-based user access control. For more information, see Controlling access to documents in an index.	November 5, 2020

New feature	The Amazon Kendra Confluence data source connector now works with Confluence cloud. For more information, see Using a Confluence data source.	November 5, 2020
Region expansion	Amazon Kendra is now available in Asia Pacific (Sydney) (ap-southeast-2).	November 2, 2020
New feature	Amazon Kendra now provides a data source connector for Confluence server. For more information, see Using a Confluence data source.	October 26, 2020
New feature	Amazon Kendra now provides a data source that you can use to generate statistics for your custom connectors. For more information, see Using a custom data source.	October 21, 2020
New feature	Amazon Kendra now supports custom attributes for frequently asked questions . For more information, see Adding questions and answers.	September 17, 2020
New feature	Amazon Kendra now returns confidence scores for query results. For more information, see QueryResultItem.	September 15, 2020

New feature	AWS CloudFormation now supports Amazon Kendra. For more information, see Amazon Kendra resource type reference - AWS CloudFormation.	September 10, 2020
New feature	Amazon Kendra adds support for AWS PrivateLink. For more information, see Amazon Kendra and interface VPC endpoints (AWS PrivateLink).	July 7, 2020
New guide	This is the first release of the *Amazon Kendra Developer Guide*.	May 11, 2020