Table of Contents

What is Amazon Q? ......................................................................................................................... 1
  Benefits of Amazon Q ................................................................................................................ 2
  Pricing and availability .............................................................................................................. 2
Accessing Amazon Q ..................................................................................................................... 3
Related services .............................................................................................................................. 3
Are you a first-time Amazon Q user? ............................................................................................ 4

Getting started with Amazon Q .................................................................................................... 5
  How Amazon Q works ................................................................................................................. 5
    Admin workflow ...................................................................................................................... 5
    User workflow ...................................................................................................................... 6
    Amazon Q workflow ............................................................................................................. 7
  Key concepts ............................................................................................................................. 7
  Document attributes and types .................................................................................................. 11
    Understanding document attributes .................................................................................. 11
    Mapping document attributes in Amazon Q ...................................................................... 15
    Supported document formats in Amazon Q ....................................................................... 17
  Supported languages ............................................................................................................... 20
  Setting up .................................................................................................................................. 20
    Initial AWS account setup ................................................................................................. 20
    (Optional) Install the AWS CLI ......................................................................................... 22
    (Optional) Set up the AWS SDKs ...................................................................................... 22
    Consider AWS Regions and endpoints .............................................................................. 23
    Set up required permissions for Amazon Q ....................................................................... 23
  IAM roles .................................................................................................................................. 24
    IAM role for an Amazon Q application ............................................................................... 25
    IAM role for an Amazon Q web experience ....................................................................... 27
    IAM role for Amazon Q data source connectors ............................................................... 28
    IAM role for Amazon Q plugins ....................................................................................... 40
    IAM roles for Lambda functions ......................................................................................... 41
    IAM role for an Amazon Kendra retriever ...................................................................... 46

Configuring an application .......................................................................................................... 48
  Creating an application ............................................................................................................. 48
  Managing applications ............................................................................................................ 50
Creating a retriever ................................................................................................................. 54
  Amazon Q retriever ............................................................................................................... 55
  Amazon Kendra retriever ...................................................................................................... 60
Connecting data sources ........................................................................................................ 65
  Upload documents .................................................................................................................. 65
  Amazon Kendra retriever ...................................................................................................... 68
  Amazon Q data sources ........................................................................................................ 69

Data source connectors ......................................................................................................... 79

  Concepts ............................................................................................................................... 79
    Source and endpoint metadata ......................................................................................... 80
    Authorization ...................................................................................................................... 80
    Authentication .................................................................................................................... 81
    Virtual private cloud ......................................................................................................... 81
    Web proxy .......................................................................................................................... 82
    IAM role .............................................................................................................................. 82
    Identity crawler .................................................................................................................. 83
    Sync scope ........................................................................................................................ 84
    Sync mode .......................................................................................................................... 84
    Sync run schedule ............................................................................................................. 85
    Field mappings ................................................................................................................... 85
Configuration best practices .................................................................................................. 86

Supported connectors .......................................................................................................... 87
  AEM (Cloud) ........................................................................................................................ 88
  AEM (Server) ......................................................................................................................... 123
  Alfresco (Cloud) .................................................................................................................. 159
  Alfresco (Server) .................................................................................................................. 177
  Aurora (MySQL) .................................................................................................................. 195
  Aurora (PostgreSQL) .......................................................................................................... 212
  Amazon FSx (Windows) ....................................................................................................... 229
  Amazon RDS (Microsoft SQL Server) ................................................................................ 243
  Amazon RDS (MySQL) ........................................................................................................ 259
  Amazon RDS (Oracle) ......................................................................................................... 276
  Amazon RDS (PostgreSQL) ............................................................................................... 294
  Amazon S3 ............................................................................................................................ 311
  Amazon Q custom connector ............................................................................................. 330
  Amazon Q Web Crawler ..................................................................................................... 334
Amazon WorkDocs ................................................................. 361
Box .................................................................................. 375
Confluence (Cloud) ................................................................ 393
Confluence (Server) .......................................................... 417
Dropbox ............................................................................. 443
Drupal ................................................................................ 461
GitHub (Cloud) ................................................................. 483
GitHub (Server) .................................................................. 506
Gmail .................................................................................. 530
Google Drive ....................................................................... 550
IBM DB2 ............................................................................. 569
Jira ..................................................................................... 586
Microsoft Exchange ........................................................... 605
Microsoft OneDrive ............................................................ 626
Microsoft SharePoint (Cloud) ............................................ 643
Microsoft SharePoint (Server) .......................................... 673
Microsoft SQL Server ...................................................... 701
Microsoft Teams .................................................................. 718
Microsoft Yammer ............................................................. 744
MySQL ................................................................................. 763
Oracle Database .............................................................. 780
PostgreSQL ......................................................................... 797
Quip .................................................................................... 814
Salesforce Online .............................................................. 830
ServiceNow Online ............................................................ 857
Slack ................................................................................... 879
Zendesk ................................................................................ 897
Understanding User Store .................................................. 918
Principal mapping ............................................................ 919
How the User Store works ................................................ 921
Using Amazon VPC ........................................................... 81
Configuring Amazon VPC .................................................. 924
Connecting to Amazon VPC .............................................. 926
Using Amazon VPC with Amazon S3 .................................. 927
Connecting to a database ................................................... 931
Troubleshooting VPC connection issues ............................... 933
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Q features</td>
<td>1040</td>
</tr>
<tr>
<td>Filtering using document attributes</td>
<td>1040</td>
</tr>
<tr>
<td>Understanding document attributes</td>
<td>1041</td>
</tr>
<tr>
<td>Mapping document attributes</td>
<td>1045</td>
</tr>
<tr>
<td>Source attribution with citations in Amazon Q</td>
<td>1047</td>
</tr>
<tr>
<td>Upload files and chat in Amazon Q</td>
<td>1048</td>
</tr>
<tr>
<td>Quick prompts in Amazon Q</td>
<td>1048</td>
</tr>
<tr>
<td>Using a web experience</td>
<td>1049</td>
</tr>
<tr>
<td>Prompts</td>
<td>1049</td>
</tr>
<tr>
<td>Engage with contextual responses</td>
<td>1050</td>
</tr>
<tr>
<td>Analyze content</td>
<td>1050</td>
</tr>
<tr>
<td>Perform actions on your behalf</td>
<td>1050</td>
</tr>
<tr>
<td>Review source citations</td>
<td>1050</td>
</tr>
<tr>
<td>Upload files and chat</td>
<td>1050</td>
</tr>
<tr>
<td>Copy responses</td>
<td>1051</td>
</tr>
<tr>
<td>Provide feedback</td>
<td>1051</td>
</tr>
<tr>
<td>Conversation management</td>
<td>1051</td>
</tr>
<tr>
<td>Tagging resources</td>
<td>1052</td>
</tr>
<tr>
<td>Using tags</td>
<td>1052</td>
</tr>
<tr>
<td>Tag restrictions</td>
<td>1053</td>
</tr>
<tr>
<td>Security</td>
<td>1054</td>
</tr>
<tr>
<td>Data protection</td>
<td>1055</td>
</tr>
<tr>
<td>Data encryption</td>
<td>1056</td>
</tr>
<tr>
<td>Key management</td>
<td>1066</td>
</tr>
<tr>
<td>Identity and access management</td>
<td>1066</td>
</tr>
<tr>
<td>Audience</td>
<td>1066</td>
</tr>
<tr>
<td>Authenticating with identities</td>
<td>1067</td>
</tr>
<tr>
<td>Managing access using policies</td>
<td>1070</td>
</tr>
<tr>
<td>How Amazon Q works with IAM</td>
<td>1073</td>
</tr>
<tr>
<td>Identity-based policy examples</td>
<td>1081</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>1088</td>
</tr>
<tr>
<td>Compliance validation</td>
<td>1090</td>
</tr>
<tr>
<td>Resilience</td>
<td>1091</td>
</tr>
<tr>
<td>Infrastructure security</td>
<td>1091</td>
</tr>
<tr>
<td>Cross-service confused deputy prevention</td>
<td>1092</td>
</tr>
<tr>
<td>Configuration and vulnerability analysis</td>
<td>1093</td>
</tr>
</tbody>
</table>
Security best practices

Apply principle of least privilege

Role-based access control (RBAC) permissions

**Monitoring**

CloudTrail logs

Amazon Q information in CloudTrail

Control plane events in CloudTrail

Data plane events in CloudTrail

Amazon Q management events in CloudTrail

Understanding Amazon Q log file entries

**Service quotas**

Supported Regions

Quotas

**API reference**

**AWS Glossary**

**Document history**
Amazon Q is in preview release and is subject to change.
What is Amazon Q (For Business Use)?

**Powered by Amazon Bedrock:** AWS implements automated abuse detection. Because Amazon Q is built on Amazon Bedrock, users can take full advantage of the controls implemented in Amazon Bedrock to enforce safety, security, and the responsible use of artificial intelligence (AI).

Amazon Q is a fully managed, generative-AI powered enterprise chat assistant that you can deploy within your organization. Amazon Q enhances employee productivity by supporting key tasks such as question-answering, knowledge discovery, writing email messages, summarizing text, drafting document outlines, and brainstorming ideas. Users ask questions of Amazon Q and get comprehensive answers that are presented in a conversational manner.

Amazon Q integrates with services such as Amazon Kendra and other supported data sources such as Amazon S3, Microsoft SharePoint, and Salesforce.

**What is Amazon Q, Your Business Expert?**

---

**Note**

This is the documentation for Amazon Q (for business use). If you are looking for documentation for Amazon Q (for AWS builder use), see the Amazon Q for AWS Builder User Guide.

**Topics**

- Benefits of Amazon Q
- Pricing and availability
- Accessing Amazon Q
- Related services
- Are you a first-time Amazon Q user?
Benefits of Amazon Q

Some of the benefits of Amazon Q include:

**Accurate and comprehensive answers**

Amazon Q generates comprehensive responses to natural language queries from users by analyzing information across all enterprise content that it has access to. It can avoid incorrect statements by confining its generated responses to existing enterprise data. Amazon Q also provides citations to the sources that it used to generate its response.

**Simple to deploy and manage**

Amazon Q undertakes the complex task of developing and managing machine learning infrastructure and models so that you can build your chat solution quickly. Amazon Q connects to your data and ingests it for processing using its pre-built connectors, document retrievers, document upload capabilities.

**Configurable and customizable**

Amazon Q provides you with the flexibility of choosing what sources should be used to respond to user queries. You can control whether the responses should only use your enterprise data, or use both enterprise data and model knowledge.

**Data and application security**

Amazon Q supports access control for your data so that the right users can access the right content. Its responses to questions are based on the content that your end user has permissions to access. You can integrate your Amazon Q web experience with any SAML 2.0 supported identity provider to manage user authentication and authorization.

**Broad connectivity**

Amazon Q offers out-of-the-box connections to multiple supported data sources. For more information about data source connectors supported by Amazon Q, see [Amazon Q data source connectors](#).

**Pricing and availability**

Amazon Q is currently in preview. For more information on pricing during the preview, see [Amazon Q pricing](#).
For a list of regions where Amazon Q is currently available, see Supported regions.

During Preview, an Amazon Q application supports only 50 end users. If you need more capacity, contact Support.

**Accessing Amazon Q**

You can access Amazon Q in the following ways in the AWS Regions that it's available in:

**AWS Management Console**

You can use the AWS Management Console—a browser-based interface to interact with AWS services—to access the Amazon Q console and resources. You can perform most Amazon Q tasks using the Amazon Q console.

**Amazon Q API**

To access Amazon Q programmatically, you can use the Amazon Q API. For more information, see the Amazon Q API Reference.

**AWS Command Line Interface**

The AWS Command Line Interface (AWS CLI) is an open source tool. You can use the AWS CLI to interact with AWS services using commands in your command line shell. If you want to build task-based scripts, using the command line can be faster and more convenient than using the console.

**SDKs**

AWS SDKs provide language APIs for AWS services to use programmatically.

**Related services**

The following are some of the other AWS services that Amazon Q integrates with:

**Amazon Kendra**

Amazon Kendra is an intelligent search service that uses natural language processing and machine learning algorithms to return specific answers from your data for end user queries. If you're already an Amazon Kendra user, you can use Amazon Kendra as a data retriever for your Amazon Q web application.
Amazon S3

Amazon S3 is an object storage service. If you're an Amazon S3 user, you can use Amazon S3 as a data source for your Amazon Q application.

Are you a first-time Amazon Q user?

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

How it works

Introduces Amazon Q components and describes how they work to create your Retrieval Augmented Generation (RAG) solution.

Key concepts

Explains key concepts and important Amazon Q terminology.

Setting up

Explains key concepts and important Amazon Q terminology and outlines how to set up Amazon Q so that you can begin creating your Amazon Q application and web experience.

Creating an application

Explains how to create the Amazon Q application that powers your Amazon Q web experience.

Configuring Amazon Q data source connectors

Configuration information for specific connectors to use with your Amazon Q web experience.
Getting started with Amazon Q

To start using Amazon Q, set up an AWS account and create the necessary AWS Identity and Access Management (IAM) users and roles. To use the AWS Command Line Interface (AWS CLI) or the AWS SDKs, you must install and configure them. After learning about Amazon Q concepts and setting up, you are ready to begin creating your application.

Topics

- How Amazon Q works
- Key concepts of Amazon Q
- Document attributes and types in Amazon Q
- Supported languages for Amazon Q
- Setting up for Amazon Q
- IAM roles for Amazon Q

How Amazon Q works

With Amazon Q, you can build an interactive chat application for your organization's end users, using a combination of your enterprise data and large language model knowledge, or enterprise data only. The following sections outline how Amazon Q works.

Topics

- Admin workflow
- User workflow
- Amazon Q workflow

Admin workflow

If you're an admin user, you create and configure an Amazon Q web experience by completing the following steps:
1. **Creating the Amazon Q application** that powers your web experience.

2. **Choosing a retriever** for the application.

3. **Connecting your data sources** to—or uploading data into—the application.

4. **Enhancing and customizing the web experience** by configuring admin-level controls, and the end user chat experience. For more information, see [Enhancing an Amazon Q application](#) and [Amazon Q features](#).

5. **Previewing and customizing your web experience** to test how it looks and works for your end users. In this step, you add a title and subtitle for your web experience, and a welcome message for your end users.

6. **Deploying your web experience** for your end users by integrating with a SAML 2.0 supported identity provider (IdP). If you're using the console, this step involves switching between your IdP console and the Amazon Q console.

**User workflow**

If you're an end user using your organization's Amazon Q web experience, you perform the following steps:

1. Navigate to your organization's Amazon Q web experience URL, and sign in with your credentials.

2. Start chatting and ask questions of your organization's Amazon Q web experience. For a list of web experience capabilities, see [Using an Amazon Q web experience](#).

3. Sometimes your question requires information that's beyond the scope of your enterprise data. Then, Amazon Q responds that it couldn't find an answer in your documents, unless your admin has allowed Amazon Q to **generate responses using model knowledge**.

4. Additionally, you can ask Amazon Q to complete **any supported follow-up tasks**—like creating a Jira ticket—that your admin has configured.

---

**Note**

During Preview, an Amazon Q application supports only 50 end users. If you need more capacity, contact [Support](#).
Amazon Q stores conversation history for 30 days and maintains conversation context after a conversation ends. Conversations can be resumed from where you left off within this 30-day period.

Amazon Q workflow

In response to an end user query during a web experience chat, Amazon Q does the following:

1. Uses the retriever chosen by the admin to select and retrieve documents that are relevant to the query, following authorization and access control.
2. Generates a response to the user query using either a combination of retrieved enterprise data and model knowledge, or only enterprise data, depending on admin configuration.
3. Returns the generated response to the end user. Amazon Q assigns a unique message ID to each answer for tracking purposes.

Key concepts of Amazon Q

This section describes the key concepts and terms related to Amazon Q.

Retrieval Augmented Generation

Retrieval Augmented Generation (RAG) is a natural language processing (NLP) technique. Using RAG, generative artificial intelligence (generative AI) is conditioned on specific documents that are retrieved from a dataset. Amazon Q has a built-in RAG system. A RAG model has the following two components:

- A retrieval component retrieves relevant documents for the user query.
- A generation component takes the query and the retrieved documents and then generates an answer to the query using a large language model.

Large language model

A large language model (LLM) is a language-based, machine learning model that's tuned to a large number (billions) of parameters or and trained on a large corpus of documents.
Retriever

A retriever pulls data from an index in real time during a conversation. Amazon Q provides retrievers for Amazon Kendra indexes and also for a native index.

Index

An index is a corpus of documents. Amazon Q supports its own index linked to its native retriever where you can add and sync documents. An index has fields that you can map your document attributes to enhance your end user's chat experience. Amazon Q creates an index for you when it creates your Amazon Q native retriever.

You can also use an Amazon Kendra index as a retriever for your generative AI application.

Data source

A data source is a document repository. If you don't have a queryable interface to retrieve data from, Amazon Q provides an index to which you can sync a data source and use for your chat application. The index can crawl and synchronize documents from the data source to an Amazon Q index at regular intervals.

Data source connector

A data source connector is a mechanism for integrating and synchronizing data from multiple repositories into one container. Amazon Q supports multiple connectors so that you can build your generative AI solution with minimal configuring. For a list of Amazon Q supported connectors, see Supported connectors. For an overview of Amazon Q connector features, see Amazon Q data source connector features.

Document

In Amazon Q, a document is a unit of data. Specific document formats supported include .csv, .docx, HTML, JSON, .pdf, plaintext, .ppt, .rtf, and .xlsx. Amazon Q supports both structured and unstructured text. For more information, see Supported document types.

Application

An Amazon Q application is the primary resource that you use to create a chat solution. To create the application, you can use either the Amazon Q console or Amazon Q API actions.

Web experience

An Amazon Q web experience is the chat interface that you create using your Amazon Q application. Then, your end users can chat with your organization's Amazon Q web experience.
You can configure and customize your Amazon Q web experience using either the Amazon Q console or the Amazon Q API.

**Guardrails and chat controls**

An Amazon Q feature that lets you define global controls and topic-level controls for your application. Using this feature, you can control what sources your application will use to generate responses from, and also control what topics it will respond to and how. For more information, see [Guardrails](#).

**Plugins**

Amazon Q includes a plugins feature that you can use to interact with third-party services such as Jira and Salesforce. With the plugins feature, you can perform actions specific to that service (like creating a ticket) from within your Amazon Q web experience chat. For more information, see [Plugins](#).

**Quick prompts**

The Amazon Q quick prompts feature helps with end user discoverability of the web experience chat features. Use this feature to prompt your end user to engage with their web experience chat in specific ways. For example, you can show the available [configured plugins](#) or inform users that they can choose to summarize their chat.

**Document attributes**

Document attributes are structural metadata associated with documents, such as document title, document type, and date and time created. Amazon Q extracts document attributes during the document ingestion process to provide customizable chat and data manipulation capabilities for your application. Amazon Q offers reserved document attributes that you can use. Or, you can create custom attributes. For more information, see [Document attributes](#), [Filtering using metadata](#), and [Custom document enrichment](#).

**Filtering using document attributes**

Filtering using document attributes is an Amazon Q feature that you can use to filter your Amazon Q chat responses for your end user. For example, if you have a document attribute associated with a data source type, you can use the attribute to mandate that chat responses only be generated from a specific data source. For more information, see [Filtering using document attributes](#).
Document enrichment

Document enrichment is an Amazon Q feature that you can use to manipulate your document content and document attributes, such as performing optical character recognition (OCR) or translation. CDE uses basic and Lambda operations. For more information see, Document attributes and types and Document enrichment.

Field mappings

An Amazon Q index has fields that help you structure data to aid the retrieval process. You can map index fields to your document attributes when you add documents directly to an index, or use a data source connector.

User Store

User Store is an Amazon Q data source connector feature that streamlines user and group management across all the data sources attached to your application. For more information about how this feature works and implementation details, see Understanding User Store.

Index units

When you use an Amazon Q native retriever for your application, you must provision data storage capacity for your index. You can provision between 1–50 units for your index. Each unit corresponds to 20,000 documents.

Identity provider

An identity provider (IdP) is a service that stores, manages, maintains, and verifies user identities for your application (in this case, Amazon Q). Some examples of IdPs are AWS Identity and Access Management (IAM, Okta, and Microsoft EntraID (formerly Azure Active Directory). To deploy your Amazon Q web experience to your end users, you must integrate your web experience with an IdP that's compliant with SAML 2.0. For more information, see Key IdP integration concepts.

Security Assertion Markup Language (SAML)

SAML is an XML-based standard for transferring user identity data between the service provider (SP)—in this case, Amazon Q—and an identity provider (IdP) such as Okta, Ping, or Microsoft EntraID. SAML supports two types of sign-in flows: Service initiated and IdP initiated. Amazon Q only supports service-initiated SAML flows and IdPs that are compliant with SAML 2.0.

Tags

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

Foundation model

A foundation model (FM) is a broad, function-based machine learning model (not specific to language systems). An FM is tuned to a large number (billions) of parameters and is trained on a large corpus of documents.

Hallucination

A hallucination, in the machine learning context, is a confident response by an AI application that isn’t justified by its training data. Think of a hallucination as instances where the response doesn’t make sense in the context of the prompt, or when the responses are out of scope with the documents provided. Amazon Q offers you the ability to minimize hallucinations by allowing your retrieval system to generate responses only from your existing enterprise data.

Document attributes and types in Amazon Q

This section outlines what document attributes are, how they work in Amazon Q, and what they can help you do for your chat solution. This section also lists the document types supported by Amazon Q.

Topics

- Understanding document attributes in Amazon Q
- Mapping document attributes in Amazon Q
- Supported document formats in Amazon Q

Understanding document attributes in Amazon Q

Every document has structural attributes—or metadata—attached to it. Document attributes can include information such as document title, document author, time created, time updated, and document type.

You can map document attributes to fields in your Amazon Q index. Once mapped to document attributes, these index fields can be used by end users to filter and scope their chat results to specific data.
### Note
Currently, filtering using document attributes in chat is only supported through the API.

You can use document attributes to prepare your data for—and customize and control—end user chat. To learn more, see [Filtering using metadata](#) and [Document enrichment in Amazon Q](#).

#### Topics
- [Reserved document attributes](#)
- [Custom document attributes](#)
- [Mapped document attributes](#)
- [Document attribute data types](#)

### Reserved document attributes
Amazon Q offers the following reserved document attributes or index fields that you can map your metadata to:

- `_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 25, 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.
- `_file_type` – The file type of the document, such as .pdf or .docx.
- `_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 25, 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.

Custom document attributes

You can also create custom attributes based on your own enterprise data. Then, you can map the custom attributes to custom index fields that you create for a more tailored end user chat experience.

For example, you can create a custom field or attribute called "Department" with the values of "HR", "Sales", and "Manufacturing". Then, you can use these fields or attributes to allow your end users to filter their chat results to documents in the "HR" department, or restrict response generation to specific data stores.

You can create up to 50 custom fields or attributes.

⚠️ Important

Once created, you can’t delete or rename any attributes.

Topics

• Ingesting attributes using the BatchPutDocument API operation
• Adding custom attributes or fields to an Amazon S3 data source

Ingesting attributes using the BatchPutDocument API operation

When you use the BatchPutDocument API operation to add a document to your index, you can specify document attributes—both reserved and custom—as part of Attributes. You can add multiple fields or attributes when you call the API operation. You can create up to 50 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 Amazon S3 bucket as a data source for your index, you use companion metadata files to add metadata to the documents. 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 50 custom fields or attributes. 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"]
}
```

Mapped document attributes

When a document attribute—reserved or custom—is mapped to an index field, you can configure it to perform the following action:

- **Search** – Allows end users the ability to search data with the specified attributes.

Document attribute data types

Document attributes—reserved or custom—can only be the data types that are shown in the following table.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Searchable</th>
<th>Filterable</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>No</td>
<td>Yes</td>
<td>DISABLED</td>
</tr>
<tr>
<td>Number</td>
<td>No</td>
<td>Yes</td>
<td>DISABLED</td>
</tr>
<tr>
<td>String</td>
<td>Yes</td>
<td>Yes</td>
<td>ENABLED</td>
</tr>
<tr>
<td>String list</td>
<td>Yes</td>
<td>Yes</td>
<td>ENABLED</td>
</tr>
</tbody>
</table>
Mapping document attributes in Amazon Q

Note
You can’t change an index field type after it has been created.

Note
Currently, filtering using document attributes in chat is only supported through the API.

Mapping document attributes from your documents to index fields is a multi-step process that depends on the document upload method you use.

Topics
• Mapping document attributes directly to index fields
• Mapping data source document attributes to index fields

Mapping document attributes directly to index fields

When you use the API, you must first map your document attributes to index fields before you can use them for filtering in chat. You use the following process to map document attributes to your index field:

1. You create an index by calling the CreateIndex API operation.
2. Then, you create index fields using the UpdateIndex operation. You use this method to map both reserved and custom document attributes to index fields.
3. Optionally, you can test and view the index fields that you’ve added by using the GetIndex operation.
4. Then, when you use the BatchPutDocument operation to ingest documents into your index, Amazon Q extracts your reserved or custom document attributes and maps them to the index fields that you have already created.

After you map document attributes directly to index fields using the API, you can select specific attributes for your end user to use for filtering chat responses. With the UpdateIndex API
operation, you add custom fields or attributes using the documentAttributeConfigurations parameter.

The following JSON example uses documentAttributeConfigurations to add a field called "Department" to the index.

```json
"DocumentmetadataConfigurationUpdates": [

  { 
    "Name": "Department",
    "Type": "STRING_VALUE"

  } 
]
```

**Mapping data source document attributes to index fields**

If you use an Amazon Q data source connector, you can map common or default document attributes attached to documents in your data source to fields in your Amazon Q index. You can use these document attributes to help your end user filter and scope chat responses.

⚠️ **Important**

Currently, filtering using data source document attributes in chat is only supported through the API.

Each data source connector is designed to crawl the default document attributes in your data source automatically. For example, if you have a field in your data source named dept that contains department information for a document, you can map it to an index field named Department. You can't change or customize default data source attributes that are mapped to an index.

You can also map any Amazon Q reserved or common fields such as _created_at. If your data source has a field named creation_date, you can map this field to the equivalent Amazon Q reserved field named _created_at.

You can also choose to add custom document attributes and map them to custom fields that you create in your index. Most data sources support field mappings and follow a specific configuration format, except Amazon S3 and database data sources:
• If you store your documents in an Amazon S3 bucket or Amazon S3 data source, you can either use the console to specify field mappings or specify fields using a JSON metadata file.

• For database data sources, if the name of the database column matches the name of a reserved field, the field and column are mapped automatically.

If you use the console, you select default field mappings or create custom mappings when you configure your connector. On the console, if a default field or a default field property can’t be edited, it will appear grayed out.

Note
Currently, you can’t use field mappings configured on the console to filter chat responses.

If you use the API, you use the configuration parameter of the `CreateDataSource` API operation to map default document attributes in your data source to index fields. If you want to map custom document attributes in your data source to Amazon Q index fields, use the `DocumentAttribute` parameter of the `UpdateIndex` operation to first create the custom field matching the custom document attribute. By doing so, you can specify and map your reserved or custom data source document attribute to a reserved or custom index field.

**Supported document formats in Amazon Q**

When you add documents to an Amazon Q application using the console or the API, Amazon Q extracts document content and metadata and internally parses these to optimize chat responses. The maximum file size of a single document must be 50 MB or less. The maximum amount of text extracted from a single document is 5 MB.

The following table shows the document formats that Amazon Q supports.

<table>
<thead>
<tr>
<th>Document format</th>
<th>Treated as</th>
<th>How document is treated</th>
<th>Original structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portable Document Format (PDF)</td>
<td>HTML</td>
<td>Converted to HTML, then content is extracted.</td>
<td>Unstructured</td>
</tr>
<tr>
<td>Document format</td>
<td>Treated as</td>
<td>How document is treated</td>
<td>Original structure</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>HyperText Markup Language (HTML)</td>
<td>HTML</td>
<td>HTML tags are filtered out to extract content. Content must between the main HTML start and closing tags (&lt;HTML&gt;content&lt;/HTML&gt;).</td>
<td>Semi-structured</td>
</tr>
<tr>
<td>Extensible Markup Language (XML)</td>
<td>XML</td>
<td>XML tags are filtered out to extract content.</td>
<td>Semi-structured</td>
</tr>
<tr>
<td>Extensible Stylesheet Language Transformations (XSLT)</td>
<td>XSLT</td>
<td>Tags are filtered out to extract content.</td>
<td>Semi-structured</td>
</tr>
<tr>
<td>Markdown (MD)</td>
<td>Plain text</td>
<td>Content is extracted with Markdown syntax included.</td>
<td>Semi-structured</td>
</tr>
<tr>
<td>Comma Separated Values (CSV)</td>
<td>CSV</td>
<td>Content extracted from each cell, with a single file treated as a single document result.</td>
<td>Structured for FAQ files; otherwise semi-structured.</td>
</tr>
<tr>
<td>Microsoft Excel (XLS and XLSX)</td>
<td>XLS and XLSX</td>
<td>Content extracted from each cell, with a single file treated as a single document result.</td>
<td>Semi-structured</td>
</tr>
<tr>
<td>Document format</td>
<td>Treated as</td>
<td>How document is treated</td>
<td>Original structure</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------------</td>
<td>---------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>JavaScript Object Notation (JSON)</td>
<td>Plain text</td>
<td>Content is extracted with JSON syntax included.</td>
<td>Semi-structured</td>
</tr>
<tr>
<td>Rich Text Format (RTF)</td>
<td>RTF</td>
<td>RTF syntax is filtered out to extract content.</td>
<td>Semi-structured</td>
</tr>
<tr>
<td>Microsoft PowerPoint (PPT)</td>
<td>PPT</td>
<td>Only text content is extracted from PowerPoint slides for ingestion. Images and other content aren't extracted.</td>
<td>Unstructured</td>
</tr>
<tr>
<td>Microsoft Word (DOCX)</td>
<td>DOCX</td>
<td>Only text content is extracted from Word pages for ingestion. Images and other content aren't extracted.</td>
<td>Unstructured</td>
</tr>
<tr>
<td>Plain text (TXT)</td>
<td>TXT</td>
<td>All text in the text document is extracted.</td>
<td>Unstructured</td>
</tr>
</tbody>
</table>

**Note**

If you use an Amazon Q connector to crawl a database, Amazon Q considers each row in a database as a single document.
Supported languages for Amazon Q

Amazon Q can respond in multiple languages. However, Amazon Q performs optimally for English language conversations and interactions. Amazon Q only indexes English language documents when you connect a Amazon Q data source or directly upload documents into your application. We recommend indexing only English language content.

Setting up for Amazon Q

Before you begin using Amazon Q for the first time, complete the following tasks.

Topics

- Initial AWS account setup
- (Optional) Install the AWS CLI
- (Optional) Set up the AWS SDKs
- Consider AWS Regions and endpoints
- Set up required permissions for Amazon Q

Initial AWS account setup

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

2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call 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 an administrative 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 an administrative user

After you sign up for an AWS account, 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 an administrative user

- For your daily administrative tasks, grant administrative access to an administrative user in AWS IAM Identity Center.

For instructions, see Getting started in the AWS IAM Identity Center User Guide.

Sign in as the administrative user

- 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.
(Optional) Install the AWS CLI

The AWS Command Line Interface (AWS CLI) is a unified developer tool for managing AWS services, including Amazon Q.

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, 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 Q, run the following commands:

   ```
   aws qbusiness help
   ```

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

(Optional) Set 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 following 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 don't 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.

To install Python, complete the following steps:

```bash
# 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.

**Consider AWS Regions and endpoints**

An endpoint is a URL that's the entry point for a web service. Each endpoint is associated with a specific AWS Region.

If you use a combination of the Amazon Q console, the AWS CLI, and the Amazon Q SDKs, pay attention to their default Regions. All Amazon Q components of a given application must be created in the same Region. Examples of a component include a retriever, an index, and a chat experience.

For the regions and endpoints supported by Amazon Q, see Regions and Endpoints.

**Set up required permissions for Amazon Q**

If you use Amazon Q through the AWS Management Console, required permissions are added on your behalf.

To use Amazon Q as an IAM user on the AWS CLI, or AWS SDK, you must attach the following permissions to allow Amazon Q to create and manage resources on your behalf:
For a complete list of IAM roles for Amazon Q, see IAM roles for Amazon Q.

IAM roles for Amazon Q

When you create an application or a web experience with Amazon Q, or connect a data source to it, Amazon Q needs access to the required AWS resources.

If you use the AWS CLI or an AWS SDK, you must create an AWS Identity and Access Management (IAM) policy before you create the Amazon Q resource. When you call the operation, you provide the Amazon Resource Name (ARN) role with the policy attached.

If you use the AWS Management Console, you can create a new IAM role in the Amazon Q console or use an existing IAM role. The console displays roles that have the string qbusiness or QBusiness in the role name.

The following topics provide details for the required policies. If you create IAM roles using the Amazon Q console, these policies are created on your behalf.

Topics

- IAM role for an Amazon Q application
- IAM role for an Amazon Q web experience
- IAM role for Amazon Q data source connectors
- IAM role for Amazon Q plugins
- IAM roles for Lambda functions
- IAM role for an Amazon Kendra retriever
IAM role for an Amazon Q application

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

To allow Amazon Q to access a CloudWatch log, use the following role policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AmazonQApplicationPutMetricDataPermission",
      "Effect": "Allow",
      "Action": ["cloudwatch:PutMetricData"],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudwatch:namespace": "AWS/QBusiness"
        }
      }
    },
    {
      "Sid": "AmazonQApplicationDescribeLogGroupsPermission",
      "Effect": "Allow",
      "Action": ["logs:DescribeLogGroups"],
      "Resource": "*"
    },
    {
      "Sid": "AmazonQApplicationCreateLogGroupPermission",
      "Effect": "Allow",
      "Action": ["logs:CreateLogGroup"],
      "Resource": [
        "arn:aws:logs:{{region}}:{{account_id}}:log-group:/aws/qbusiness/*"
      ]
    },
    {
      ...
    }
  ]
}
```
To allow Amazon Q to assume a role, use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AmazonQApplicationPermission",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{account_id}}"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{account_id}}:application/*"
        }
      }
    }
  ]
}
```
IAM role for an Amazon Q web experience

When you deploy an Amazon Q web experience, you must provide Amazon Q with an IAM role with permissions to write access relevant Amazon Q API operations. You must also provide a trust policy that allows Amazon Q to assume the role. The following are the policies that must be provided.

To allow Amazon Q to access the API operations required to deploy your web experience, use the following role policy:

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Sid": "QBBusinessConversationPermission",
      "Effect": "Allow",
      "Action": [
         "qbusiness:Chat",
         "qbusiness:ChatSync",
         "qbusiness:ListMessages",
         "qbusiness:ListConversations",
         "qbusiness:GetWebExperience",
         "qbusiness:GetApplication",
         "qbusiness:PutFeedback",
         "qbusiness:DeleteConversation"
      ],
      "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
   }
   ]
}
```

To allow Amazon Q to assume a role, use the following trust policy:

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Sid": "QBBusinessTrustPolicy",
      "Effect": "Allow",
      "Principal": {
         "Service": "application.qbusiness.amazonaws.com"
      }
   }
   ]
}
```
IAM role for Amazon Q data source connectors

You can use either the Amazon Q console or the CreateDataSource API operation to connect your data source. However, you must first provide Amazon Q with an IAM role that has permissions to access the data source resources.

If you use the console, you can either create an IAM role when you connect your data source to Amazon Q or use an existing role. If you use the CreateDataSource API operation, you must provide the Amazon Resource Name (ARN) of an existing IAM role.

The specific permissions required depend on the data source. At a minimum, your IAM role must include the following:

- Permission to access the BatchPutDocument and BatchDeleteDocument API operations in order to ingest documents.
- Permission to access the User Store APIs needed to ingest access control and identity information from documents.

To allow Amazon Q to connect to your data source, use the following least-permissions role policy:

```
"Action": "sts:AssumeRole",
"Condition": {
  "StringEquals": {
    "aws:SourceAccount": "{{source_account}}"
  },
  "ArnEquals": {
    "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
  }
}
```

Note
This policy assumes your data source doesn't use any authentication.
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
      "Effect": "Allow",
      "Principal": { ...
```
If your data source uses authentication, you must add the following policy to your IAM role to allow Amazon Q to access your AWS Secrets Manager secret:

```json
{
    "Sid": "AllowsAmazonQToGetSecret",
    "Effect": "Allow",
    "Action": [
        "secretsmanager:GetSecretValue"
    ],
    "Resource": [
        "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:{{secret_id}}"
    ]
}
```

If you are using an Amazon VPC, you must add the following VPC access permissions to your policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToCreateAndDeleteNI",
            "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]]"
],
{
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
    "StringLike": {
        "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
        "aws:TagKeys": [
            "AMAZON_Q"
        ]
    }
}
},
{
"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
    "ec2:CreateTags"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
    "StringEquals": {
        "ec2:CreateAction": "CreateNetworkInterface"
    }
}
},
{
"Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
"Effect": "Allow",
"Action": [
    "ec2:CreateNetworkInterfacePermission"
]}
}
If your Secrets Manager secret is decrypted, you must add permissions for AWS KMS key to decrypt the username and password secret stored by Secrets Manager:

```json
{
  "Effect": "Allow",
  "Action": [
    "kms:Decrypt"
  ],
  "Resource": [
    "arn:aws:kms:your-region:your-account-id:key/key-id"
  ],
  "Condition": {
    "StringLike": {
      "kms:ViaService": [
        "secretsmanager.*.amazonaws.com"
      ]
    }
  }
}
```
If your Amazon Q data source connector needs access to an object stored in an Amazon S3 bucket (such as an SSL certificate), you must add the following permissions to your IAM role:

```
{
    "Sid": "AllowsAmazonQToGetS3Objects",
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::{{input_bucket_name}}/*"
    ],
    "Effect": "Allow",
    "Condition": {
        "StringEquals": {
            "aws:ResourceAccount": "{{account_id}}"
        }
    }
}
```

**Note**

Check that the file path to the object in your Amazon S3 bucket is of the following format: `s3://BucketName/FolderName/FileName.extension`.

---

**IAM role for Amazon S3 data sources**

When you use an Amazon S3 bucket as a data source, you must provide a role that has permissions to:

- Access your Amazon S3 bucket.
- Permission to access the [BatchPutDocument](https://docs.aws.amazon.com/qLiveData/docs/batch-pu.html) and [BatchDeleteDocument](https://docs.aws.amazon.com/qLiveData/docs/batch-del.html) API operations in order to ingest documents.
- Permission to access the Principal Store APIs needed to ingest access control and identity information from documents.
Note

Amazon Q doesn't use a bucket policy that grants permissions to an Amazon Q principal to interact with an Amazon S3 bucket. Instead, Amazon Q uses IAM roles. To avoid any data security issues in accidentally granting permissions to arbitrary principals, make sure that Amazon Q isn't included as a trusted member in your bucket policy. However, you can add a bucket policy to use an Amazon S3 bucket across different AWS accounts.

To allow Amazon Q to use an Amazon S3 bucket as a data source, use the following role policy:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowsAmazonQToGetObjectfromS3",
         "Action": ["s3:GetObject"],
         "Resource": ["arn:aws:s3:::{{input_bucket_name}}/*"],
         "Effect": "Allow",
         "Condition": {
            "StringEquals": {
               "aws:ResourceAccount": "{{account_id}}"
            }
         }
      },
      {
         "Sid": "AllowsAmazonQToListS3Buckets",
         "Action": ["s3:ListBucket"],
         "Resource": ["arn:aws:s3:::{{input_bucket_name}}"],
         "Effect": "Allow",
         "Condition": {
            "StringEquals": {
               "aws:ResourceAccount": "{{account_id}}"
            }
         }
      }
   ]
}
```
If the documents in the Amazon S3 bucket are encrypted, you must provide the following permissions to use the AWS KMS key to decrypt the documents:

```json
{
    "Sid": "AllowsAmazonQToDecryptSecret",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:{{region}}:{{account_id}}:key/{{kms_key_id}}"
    ]
}
```
"arn:aws:kms:{{region}}:{{account_id}}:key/[[key_id]]",
"Condition": {
  "StringLike": {
    "kms:ViaService": [
      "secretsmanager.*.amazonaws.com"
    ]
  }
}
}

If you are using an Amazon VPC, you must add the following VPC access permissions to your policy:

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToGetObjectfromS3",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::{{input_bucket_name}}/*"
      ],
      "Effect": "Allow",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "{{account_id}}"
        }
      }
    },
    {
      "Sid": "AllowsAmazonQToListS3Buckets",
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::{{input_bucket_name}}"
      ],
      "Effect": "Allow",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "{{account_id}}"
        }
      }
    }
  ]
}
"aws:ResourceAccount": "{{account_id}}"

},

{
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ],
    "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},

{
    "Sid": "AllowsAmazonQToCallPrincipalMappingAPIs",
    "Effect": "Allow",
    "Action": [
        "qbusiness:PutGroup",
        "qbusiness:CreateUser",
        "qbusiness:DeleteGroup",
        "qbusiness:UpdateUser",
        "qbusiness:ListGroups"
    ],
    "Resource": [
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**"
    ]
},

{
    "Sid": "AllowsAmazonQToCreateAndDeleteENI",
    "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]]"
    ]
}
IAM role for Amazon Q data source connectors

```json
{
  "Sid": "AllowsAmazonQToCreateDeleteENI",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "AMAZON_Q"
      ]
    }
  }
},
{
  "Sid": "AllowsAmazonQToCreateTags",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
    }
  }
},
{
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    }
  }
}
```
To allow Amazon Q to assume a role, use the following trust policy:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
         "Effect": "Allow",
         "Principal": {
            "Service": "qbusiness.amazonaws.com"
         },
         "Action": "sts:AssumeRole",
         "Condition": {
            "StringEquals": {
               "aws:SourceAccount": "{{source_account}}"
            },
            "ArnLike": {
               "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
            }
         }
      }
   ]
}
```
IAM role for Amazon Q plugins

To successfully connect Amazon Q to a plugin, you need to give Amazon Q the following permissions using a service access role:

- Permission to access your Secrets Manager secret to get the credentials you use to log in to the third party service instance you are creating a plugin for.
- **(Optional)** Permission to access the customer managed AWS KMS key used to encrypt the content of your Secrets Manager secret.

Amazon Q assumes this role to access your third party service instance credentials.

If you use the console and choose to create a new IAM role, Amazon Q creates the IAM role for you. If you use the console and choose to use an existing secret, or you use the API, make sure your secret contains the following permissions.

The following is the service access IAM role required:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowQBusinessToGetSecretValue",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[{secret_id}]"
            ]
        }
    ]
}
```

To allow Amazon Q to assume a role, use the following trust policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
```
IAM roles for Lambda functions

Custom document enrichment (CDE) is an Amazon Q feature that you can use to manipulate your document content and document attributes. When you use the Lambda functions for CDE, you need an IAM role for the following:

- A role for PreExtractionHookConfiguration with permissions to run PreExtractionHookConfiguration and to access the Amazon S3 bucket when you use PreExtractionHookConfiguration.
- A role for PostExtractionHookConfiguration with permissions to run PostExtractionHookConfiguration and to access the Amazon S3 bucket when you use PostExtractionHookConfiguration.

Both AWS Identity and Access Management (IAM) roles must have the permissions to:

- Run PreExtractionHookConfiguration and/or PostExtractionHookConfiguration. To apply advanced alterations of your document metadata and content during the ingestion process, configure a Lambda function for PreExtractionHookConfiguration and/or PostExtractionHookConfiguration.
• (Optional) If you choose to activate Server Side Encryption for your Amazon S3 bucket, you must provide permissions to use the AWS KMS key customer to encrypt and decrypt the objects stored in your Amazon S3 bucket.

A role policy to allow Amazon Q to run PreExtractionHookConfiguration with encryption for your Amazon S3 bucket.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "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:your-region:your-account-id:key/key-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
An role policy to allow Amazon Q to run PreExtractionHookConfiguration without encryption.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "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:your-region:your-account-id:function:pre-extraction-lambda-function"
        }
    ]
}
```
A role policy to allow Amazon Q to run 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",
                "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:your-region:your-account-id:key/key-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": "arn:aws:lambda:your-region:your-account-id:function:post-extraction-lambda-function"
        }
    ]
}
```
An role policy to allow Amazon Q to run PostExtractionHookConfiguration without encryption.

```json
{
    "Version": "2012-10-17",
    "Statement": [{
        "Action": [
            "s3:GetObject",
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::bucket-name",
            "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:your-region:your-account-id:function:post-extraction-lambda-function"
    }
}
```

We recommend that you include `aws:sourceAccount` and `aws:sourceArn` in the trust policy. Their inclusion limits permissions and securely checks if `aws:sourceAccount` and `aws:sourceArn` are the same values as provided in the IAM role policy for the `sts:AssumeRole`
This approach prevents unauthorized entities from accessing your IAM roles and their permissions. For more information, see confused deputy problem in the IAM User Guide.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "Service": ["qbusiness.amazonaws.com"]
         },
         "Action": "sts:AssumeRole",
         "Condition": {
            "StringEquals": {
               "aws:SourceAccount": "your-account-id"
            },
            "StringLike": {
               "aws:SourceArn": "arn:aws:qbusiness:your-region:your-account-id:index-id/**"
            }
         }
      }
   ]
}
```

IAM role for an Amazon Kendra retriever

When you use an Amazon Kendra index as a retriever, you must provide Amazon Q with an IAM role with permissions to access Amazon Kendra. You must also provide a trust policy that allows Amazon Q to assume the role. The following are the policies that must be provided.

To allow Amazon Q to access a CloudWatch log, use the following policy:

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "KendraRetrieveAccess",
         "Effect": "Allow",
         "Action": ["cloudwatch:PutLogEvents"]
      }
   ]
}
```
To allow Amazon Q to assume a role, use the following trust policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AmazonQKendraAccessPermission",
            "Effect": "Allow",
            "Principal": {
                "Service": "qbusiness.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "{{source_account}}"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{applicationId}}"
                }
            }
        }
    ]
}
```
Configuring an Amazon Q application

As the first step towards creating an Amazon Q chat application for your end users, you configure an Amazon Q application. Then, you can optionally enhance it by customizing the end user experience. After this, you select and create a retriever, and connect and configure the data sources.

This section guides you through the process of creating and configuring an Amazon Q application. To create an application, you can use the Amazon Q console, the AWS Command Line Interface (AWS CLI), and the Amazon Q API operations.

As a prerequisite, make sure that you complete the setting up tasks. If you're using the AWS CLI or the API, make sure that you created the required IAM roles.

After you finish creating your application, you can customize and preview the web experience that it will power.

⚠️ Note

During Preview, an Amazon Q application supports only 50 end users. If you need more capacity, contact Support.

Topics

- Creating an Amazon Q application
- Creating and selecting a retriever for an Amazon Q application
- Connecting data sources to an Amazon Q application

Creating an Amazon Q application

To create an Amazon Q application, you can use either the AWS Management Console or the Amazon Q API.

Before you begin to create an Amazon Q application, make sure that you complete the setting up tasks. If you're using the AWS CLI or the Amazon Q API, make sure that you created the required IAM roles.

After you create an application, you can create your Amazon Q web experience. How you create the web experience depends on whether you use the AWS Management Console or the Amazon Q APIs.
• **AWS Management Console** – If you use the console to create an application, the web experience is created automatically.

• **Amazon Q API** – If you use the `CreateApplication` API operation to create an application, use the `CreateWebExperience` API operation to create your web experience.

The following tabs provide a procedure for creating an application by using the AWS Management Console and code examples for using the AWS CLI.

**Console**

**To configure an Amazon Q application**

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. For **Create Amazon Q application**, choose **Get started**.
3. For **Applications**, choose **Create application**.
4. For **Application settings**, enter the following information for your Amazon Q application:
   - **Application name** – A name for your Amazon Q application for easy identification. This name is only visible in the AWS Management Console. The name can include hyphens (-), but not spaces, and can have a maximum of 1,000 alphanumeric characters.
   - **Service access** – An IAM role for Amazon Q to allow it to access the AWS resources it needs to create your application. You can choose to use an existing role or create a new role.
     - **Note**
     For more information about example service roles, see [IAM role for an Amazon Q application](#).
   - **Service role name** – A name for the service (IAM) role you created for easy identification on the console.
   - **Encryption** – Amazon Q encrypts your data by default using AWS managed AWS KMS keys. To customize your encryption settings, select **Customize encryption settings (advanced)**. Then, you can choose to use an existing AWS KMS key or create a new one. To learn more, see [Data encryption](#).
Important

If you choose to use a customer managed key, you must provision at least 10 index storage units when you create an Amazon Q retriever.

5. **Tags – optional** – To add tags to your Amazon Q application and web experience, select Add new tag. Then, enter the following information for each tag:

   - **Key** – Add a key for your tag.
   - **Value - optional** – An optional value for your tag.

For more information about using tags with Amazon Q, see Tags.

6. To start creating your application, choose Create.

AWS CLI

To configure an Amazon Q application

```
aws qbusiness create-application \
--display-name application-name \
--role-arn roleArn \
--description optional-app-description \
--encryption-configuration kmsKeyId=<kms-key-id> \
--attachments-configuration attachmentsControlMode=ENABLED \
```

Managing Amazon Q applications

To manage an Amazon Q application, you can take the following actions:

**Actions**

- Deleting an application
- Getting application properties
- Listing applications
- Updating an application
Deleting an application

To delete an Amazon Q application, you can use the console or the `DeleteApplication` API operation.

The following tabs provide a procedure for the console and code examples for the AWS CLI.

Console

To delete an Amazon Q application

1. Sign in to the AWS Management Console and open the Amazon Q console at [https://console.aws.amazon.com/amazonq/](https://console.aws.amazon.com/amazonq/).
2. For Applications, choose Actions.
3. Choose Delete.
4. In the dialog box that opens, type Delete to confirm deletion, and then choose Delete.

You are returned to the service console while your application is deleted. When the deletion process is complete, the console displays a message confirming successful deletion.

AWS CLI

To delete an Amazon Q application

```
aws qbusiness delete-application \
--application-id application-id
```

Getting application properties

To get the properties of an Amazon Q application, you can use the console or the `GetApplication` API operation.

The following tabs provide a procedure for the console and code examples for the AWS CLI.

Console

To get properties of an Amazon Q application
1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.

2. For Applications, select the name of your application from the list of applications.

3. On Application settings, the following properties are available:
   - **Application name** – The name that you chose for your application.
   - **Application ID** – The ID assigned to your application.
   - **Subtitle** – The subtitle that you chose to assign to your application.
   - **Service access** – The service access role that your application is using.
   - **Title** – The title that you gave to your application.
   - **Application status** – The status of your application.

   To update a setting, select Edit.

AWS CLI

**To get Amazon Q application properties**

```bash
aws qbusiness get-application \
--application-id application-id
```

**Listing applications**

To list Amazon Q applications, you can use the console or the ListApplications API operation.

The following tabs provide a procedure for the console and code examples for the AWS CLI.

**Console**

**To list your Amazon Q applications**

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.

2. In Applications, all your configured applications are listed.
AWS CLI

**To list Amazon Q applications**

```bash
aws qbusiness list-applications \
--application-id application-id \
--max-results max-results-to-return
```

**Updating an application**

To update an Amazon Q application, you can use the console or the `UpdateApplication` API operation.

The following tabs provide a procedure for the console and code examples for the AWS CLI.

**Console**

**To update an Amazon Q application**

**Option 1**

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In **Applications**, select the name of your application from the list of applications.
3. In **Applications**, choose **Actions**.
4. Choose **Edit**.
   
   On the **Update application** page, edit your application settings.

**Option 2**

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In **Applications**, select the name of your application from the list of applications.
3. On your application page, select **Edit** from the page header, or select **Edit** from **Application settings**.
4. Choose **Edit**.
On the **Update application** page, edit your application settings.

### AWS CLI

**To update an Amazon Q application**

```
aws qbusiness update-application \
--application-id application-id \
--display-name application-name \
--role-arn optional-roleArn \
--description optional-app-description \
--attachments-configuration attachmentsControlMode=ENABLED
```

### Creating and selecting a retriever for an Amazon Q application

After creating your Amazon Q application, you create and select the retriever that will power your generative AI web experience. You can choose between selecting an Amazon Q retriever or using an already configured Amazon Kendra index as a retriever.

To select a retriever, you use the AWS Management Console or the [CreateRetriever](#) API operation.

If you use the console and choose to use an Amazon Q retriever, Amazon Q creates an index for you as part of the application configuration process. For easy tracking, you can tag both the retriever and index. If you use the API to create an Amazon Q retriever, you must also use the [CreateIndex](#) API operation to create an Amazon Q index.

⚠️ **Important**

You can't change the retriever for your application after your application has been created.

To change your retriever, you must create a new application.

**Note**

The data sources available to connect to your application change depending on your retriever choice.
For instructions on how to select a retriever, choose a topic based on your retriever preference for Amazon Q.

**Topics**
- Creating an Amazon Q retriever
- Selecting an Amazon Kendra retriever to an Amazon Q application

**Creating an Amazon Q retriever**

To select a Amazon Q retriever, you can use either the AWS Management Console, or the CreateIndex and CreateRetriever API operations.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

**Console**

**To create an Amazon Q retriever**

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. Complete the steps to create your Amazon Q application.
3. Then, for Select retriever, choose Use native retriever – Build an Amazon Q retriever for your Amazon Q application. This option creates an Amazon Q index that can connect to the Amazon Q supported data sources that you choose.

**Important**
The native retriever includes a default capacity of 10k documents and 0.5 queries per second (QPS).

**Note**
Available data sources when you select this option include all Amazon Q supported data connectors and direct document upload.
4. For **Index provisioning** – Choose the **Number of units** that you need. Amazon Q charges you based on the document capacity that you choose. You can choose up to 50 units. Each unit contains 20,000 documents. You can provision up to 10,000,000 documents.

5. For **Tags** – Choose whether you want to add **Index tags**.

6. To create your retriever, choose **Create**.

**AWS CLI**

**To create an Amazon Q index**

```bash
aws qbusiness create-index \
--application-id application-id \
--display-name display-name \
--description optional-index-description \
--capacity-configuration-units <index-capacity-units>
```

**To create an Amazon Q retriever**

```bash
aws qbusiness create-retriever \
--application-id application-id \
--display-name display-name \
--type NATIVE_INDEX \
--role-arn optional-roleArn \
--configuration nativeIndexConfiguration="{indexId=<created-index-id}>" \
--tags optional-tags
```

**Managing Amazon Q retrievers**

To manage Amazon Q retrievers, you can take the following actions:

**Actions**

- [Deleting an Amazon Q retriever](#)
- [Getting properties of an Amazon Q retriever](#)
- [Listing Amazon Q retrievers](#)
- [Updating Amazon Q retrievers](#)
Deleting an Amazon Q retriever

To delete a Amazon Q retriever and its associated index, you can use the console or the
DeleteRetriever API operation.

If you use the DeleteIndex API operation, deleting a retriever also deletes the Amazon Q index
that's attached to it. You can't selectively choose to delete an index attached to a retriever.

If you're using the console, the only way to delete your Amazon Q native retriever and the index
associated with it, is to delete your Amazon Q application.

The following tabs provide a procedure for the AWS Management Console and code examples for
the AWS CLI.

Console

To delete an Amazon Q retriever

1. Sign in to the AWS Management Console and open the Amazon Q console at https://
   console.aws.amazon.com/amazonq/.
2. In Applications, choose Actions.
3. Choose Delete.
4. In the dialog box that opens, type Delete to confirm deletion, and then choose Delete.
   You are returned to the service console while your application is deleted. When the deletion
   process is complete, the console displays a message confirming successful deletion.

AWS CLI

To delete an Amazon Q retriever

`aws qbusiness delete-retriever \
--application-id application-id \
--retriever-id retriever-id`

Getting properties of an Amazon Q retriever

To get the properties of an Amazon Q retriever and index, you can use the console or the
GetRetriever API operation.
The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

**Console**

**To get properties of an Amazon Q retriever**

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In Applications, select the name of your application from the list of applications.
3. For Retriever settings, the following settings are available:
   - **Retriever** – The type of retriever that you're using.
   - **Document count** – The number of documents that are attached to your index.
   - **Last modified time** – The time that your index was last modified.
   - **Index ID** – The ID of the index attached to your retriever.
   - **Storage used** – The amount of storage that your index is using.
   - **Index status** – The status of your index.

**AWS CLI**

**To get properties of an Amazon Q retriever**

```
aws qbusiness get-retriever \ 
--application-id application-id \ 
--retriever-id retriever-id
```

**Listing Amazon Q retrievers**

To list your native Amazon Q retrievers, you can use the console or the ListRetrievers API operation.
If you use the console, the list of Amazon Q retrievers and indices attached to them correspond to the list of applications that you have created.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

Console

To list your Amazon Q retrievers

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. For Applications, a list of all retrievers (with indices associated) that you have created is available.

AWS CLI

To list your Amazon Q retrievers

```sh
aws qbusiness list-retrievers \
--application-id application-id \
--max-results maximum-result-to-display
```

Updating Amazon Q retrievers

To update your Amazon Q retriever, you can use the UpdateRetriever API operation.

You can't update your retriever and its associated index by using the console.

The following tab provides code examples for the AWS CLI.

Console

This action is not supported on the console.

AWS CLI

To update your Amazon Q retriever

```sh
aws qbusiness update-retriever \
```
Selecting an Amazon Kendra retriever to an Amazon Q application

To select an existing Amazon Kendra retriever to your Amazon Q application, you can use the AWS Management Console or the CreateRetriever API operation.

If you use the API, you select and connect your Amazon Kendra retriever when you use the CreateRetriever API operation.

If you use the console, selecting and connecting an Amazon Kendra retriever is a two-step process. This topic provides instructions for the first step: Selecting an Amazon Kendra retriever. For instructions for the second step, see Selecting an Amazon Kendra retriever to an Amazon Q application.

Note

If you use an Amazon Kendra retriever, data in your Amazon Kendra will be connected to your Amazon Q application. If you choose this option, you can't use Amazon Q data connectors or direct document upload for your application.

For more information about Amazon Kendra, see the following topics in the Amazon Kendra Developer Guide and API Reference:

- What is Amazon Kendra?
- Creating a data source connector
- Amazon Kendra API Reference

The following tabs provide a procedure for the AWS Management Console and code samples for the AWS CLI.

Console

To create an Amazon Kendra retriever
1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.

2. Complete the steps to create your Amazon Q application.

3. The, in **Select retriever** choose **Use existing retriever** – Choose an Amazon Kendra index you have previously created as a retriever. All data sources synced to your Amazon Kendra index will be connected to your Amazon Q application.

4. In **Tags** – Choose whether you want to add **Retriever tags**.

5. To connect your application to your data sources, choose **Next**.

**AWS CLI**

**To create an Amazon Kendra retriever**

```
aws qbusiness create-retriever \
--display-name display-name \
--type KENDRA_INDEX \
--role-arn roleArn \
--configuration kendraIndexConfiguration="indexId=<kendra-index-id>"
```

**Managing Amazon Kendra retrievers**

To manage Amazon Kendra retrievers, you can take the following actions:

**Actions**

- **Deleting an Amazon Kendra retrievers**
- **Getting properties of an Amazon Kendra retriever**
- **Listing Amazon Kendra retrievers**
- **Updating an Amazon Kendra retriever**

**Deleting an Amazon Kendra retrievers**

To delete an Amazon Kendra retriever, you can use the console or the [DeleteRetriever](#) API operation.

If you use the console, the only way to delete your Amazon Kendra retriever from your Amazon Q application is to delete your Amazon Q application.
The following tabs provide a procedure for the AWS Management Console and code examples for
the AWS CLI.

Console

**To delete an Amazon Kendra retriever**

1. Sign in to the AWS Management Console and open the Amazon Q console at [https://console.aws.amazon.com/amazonq/](https://console.aws.amazon.com/amazonq/).
2. In **Applications**, choose **Actions**.
3. Choose **Delete**.
4. In the dialog box that opens, type **Delete** to confirm deletion, and then choose **Delete**.

You are returned to the service console while your application is deleted. When the deletion
process is complete, the console displays a message confirming successful deletion.

AWS CLI

**To delete an Amazon Kendra retriever**

```bash
aws qbusiness delete-retriever \
  --application-id application-id \
  --retriever-id retriever-id
```

**Getting properties of an Amazon Kendra retriever**

To get the properties of an Amazon Kendra retriever, you can use the console or the [GetRetriever](https://aws.amazon.com/documentation/qbusiness/)

API operation.

The following tabs provide a procedure for the AWS Management Console and code examples for
the AWS CLI.

Console

**To get the properties of an Amazon Kendra retriever**

1. Sign in to the AWS Management Console and open the Amazon Q console at [https://console.aws.amazon.com/amazonq/](https://console.aws.amazon.com/amazonq/).
2. In **Applications**, select the name of your application from the list of applications.

3. For **Retriever settings**, the following settings are available:
   - **Retriever** – The type of retriever that you’re using.
   - **Document count** – The number of documents that are attached to your index.
   - **Last modified time** – The time that your index was last modified.
   - **Index ID** – The ID of the index attached to your retriever.
   - **Storage used** – The amount of storage that your index is using.
   - **Index status** – The status of your index.

   ![Note]
   You can't edit or update retriever or index settings.

AWS CLI

**To get properties of an Amazon Kendra retriever**

```
aws qbusiness get-retriever \
  --application-id application-id \
  --retriever-id retriever-id
```

**Listing Amazon Kendra retrievers**

To list Amazon Kendra retrievers, you can use the console or the [ListRetrievers](https://aws.amazon.com) API operation.

If you use the console, the list of native retrievers and indices attached to them correspond to the list of applications that you have created.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

**Console**

**To list Amazon Kendra retrievers**
1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.

2. For Applications, a list of all retrievers (with indices associated) that you have created is available.

**AWS CLI**

**To list Amazon Kendra retrievers**

```bash
aws qbusiness list-retrievers \
  --application-id application-id \
  --max-results maximum-result-to-display
```

**Updating an Amazon Kendra retriever**

To update your Amazon Kendra retriever, you can use the UpdateRetriever API operation.

You can't update your Amazon Kendra retriever using the console.

The following tab provides code examples for the AWS CLI.

**Console**

**This action is not supported on the console.**

**AWS CLI**

**To update an Amazon Kendra retriever**

```bash
aws qbusiness update-retriever \
  --application-id application-id \
  --retriever-id retriever-id \
  --display-name display-name \
  --role-arn roleArn \
  --configuration "kendraIndexConfiguration=\"[indexId=<kendra-index-d>]\""
```
Connecting data sources to an Amazon Q application

After you select a retriever for your Amazon Q application, you connect data sources to it. Available data sources vary based on your choice of the retriever.

If you use an Amazon Q retriever, you can choose from the following options:

- Connect to any Amazon Q supported data source connectors by using the `CreateDataSource` API operation.
- Upload documents directly by using the `BatchPutDocument` API operation.

If you use an existing Amazon Kendra retriever, only data sources already connected to your Amazon Kendra index are available in your application.

To connect data sources, choose a topic based on your data source preference for your Amazon Q application.

Topics

- Upload documents
- Amazon Kendra retriever
- Amazon Q data source connectors

Upload documents

To upload documents directly to an Amazon Q application, you can use the AWS Management Console or the `BatchPutDocument` API operation.

If you use an Amazon Kendra index to retrieve your documents, you can't directly upload documents.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

Console

To upload documents
1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.

2. Complete the steps to create your Amazon Q application.

3. Complete the steps for selecting an Amazon Q retriever.

4. Then, for Upload documents, select one of the following methods to add your files:
   - Drag and drop the document files that you want to upload.
   - Add your documents to the application, and then select Choose files.

5. After choosing your files, choose Upload.
   You are returned to the Amazon Q console while your documents are uploaded. The console displays a confirmation message when your documents are successfully uploaded.

**Note**
Files can only be uploaded after the Amazon Q retriever and index creation process has completed.

AWS CLI

To upload documents directly

```
aws qbusiness batch-put-document \
--application-id application-id \
--index-id index-id \
--documents documents-to-add \
--data-source-sync-id optional-data-source-sync-id \
--role-arn optional-roleArn
```
Delete uploaded documents

To delete documents that have been directly uploaded to an application, you can use the console or the `BatchDeleteDocument` API operation. You can delete specific documents or all documents.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

Console

**To delete specific directly uploaded documents**

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In Applications, select the name of the application that your uploaded files belong to.
3. From your applications page, from Data sources, choose Uploaded files.
4. In Uploaded files, choose Document name, and then select the documents that you want to delete.
5. Choose Delete files.

You are returned to the service console while your application is deleted. When the deletion process is complete, the console displays a message confirming successful deletion.

**To delete all directly uploaded documents**

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In Applications, select the name of the application that your uploaded files belong to.
3. From your applications page, from Data sources, select Uploaded files.
4. Select Actions, and then choose Delete.
5. When the deletion process is complete, the console displays a message confirming successful file deletion.

AWS CLI

**To delete documents**
Connecting an Amazon Kendra retriever to an Amazon Q application

To use an Amazon Kendra index as a retriever for Amazon Q, you must have already configured an Amazon Kendra index and connected it with data. For more information, see What is Amazon Kendra? and Are you a first-time Amazon Kendra user? in the Amazon Kendra Developer Guide.

To add an existing Amazon Kendra retriever to your Amazon Q application, you can use the AWS Management Console or the CreateRetriever API operation. If you use the console, selecting and connecting an Amazon Kendra retriever is a two-step process. The first step is when you select an Amazon Kendra retriever. In this topic, you perform the second step—connecting an Amazon Kendra retriever.

If you use the API, you create your web experience after connecting your Amazon Kendra retriever using the CreateWebExperience API operation. If you use the console, connecting your Amazon Kendra retriever also automatically creates your Amazon Q web experience. At the end of the retriever connection process, your Amazon Kendra powered Amazon Q web experience is ready to be previewed, enhanced, and deployed.

⚠️ Note

If you select an Amazon Kendra retriever, data in your Amazon Kendra is connected to your Amazon Q application.

Console

To connect an Amazon Kendra retriever

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. Complete the steps to create your Amazon Q application.
3. Complete the steps for selecting an Amazon Kendra retriever.

4. Then, in Content sources, for Amazon Kendra indexes – Choose the Amazon Kendra index that you want to use for your Amazon Q application. Then, enter the following information:

   - **Service access** – Provide the IAM access role to connect Amazon Kendra to Amazon Q. Use an existing role, or create a new one.
   - **Service role name** – Provide a name for your IAM access role. Or, choose to use the auto-generated role that's provided.

5. To connect your Amazon Kendra indexes to the application, choose Create application.

You are returned to the Amazon Q console while your web application is created.

AWS CLI

To create and connect an Amazon Kendra retriever

```bash
aws qbusiness create-retriever \
   --application-id application-id \
   --display-name display-name \
   --type KENDRA_INDEX \
   --role-arn roleArn \
   --configuration kendraIndexConfiguration="{indexId=<kendra-index-id>}" 
```

ɪ Note

For information on managing your Amazon Kendra retriever, see Managing Amazon Kendra retrievers.

Amazon Q data sources

To connect a data source to your Amazon Q application, you can use the AWS Management Console or the `CreateDataSource` API operation.

By using the `CreateDataSource` API operation, you can configure tags, sync run schedules, and configure Amazon VPC settings. Then, you can use the `configuration` parameter to provide all other configuration information specific to your data source connector.
If you use the console, creating the data source and configuring it are a single step. After your data source is successfully configured and added, Amazon Q automatically creates a Amazon Q web experience for you.

If you use the API, you use the `CreateWebExperience` API operation after connecting your data sources to create your web experience.

**Note**

This procedure is available if you chose the *Use native retriever* option to configure your application.

## Console

**To connect a data source to an Amazon Q application**

1. Sign in to the AWS Management Console and open the Amazon Q console at [https://console.aws.amazon.com/amazonq/](https://console.aws.amazon.com/amazonq/).
2. Complete the steps to [create your Amazon Q application](#).
3. Complete the steps for [selecting an Amazon Q retriever](#).
4. Then, from **Data sources** – Add an available data source to connect your Amazon Q application.
   
   You can add up to 5 data sources.
5. For information on configuring your chosen data source, see [Supported connectors](#) to find configuration information specific to your data source.
6. To connect your configured data source to your application, choose **Add data sources**.

At the end of this step, your Amazon Q web experience is ready to be previewed, enhanced, and deployed.

## AWS CLI

**To connect a data source**

```shell
aws qbusiness create-data-source \ 
--application-id application-id \
```
Managing Amazon Q data sources

To manage data source connectors, you can perform the following actions:

Actions

- Deleting an Amazon Q data source connector
- Getting properties of an Amazon Q data source connector
- Listing Amazon Q data source connectors
- Updating Amazon Q data source connectors
- Starting data source connector sync jobs
- Stopping data source connector sync jobs
- Listing data source connector sync jobs

Deleting an Amazon Q data source connector

To delete an Amazon Q data source connector, you can use the console or the `DeleteDataSource` API operation.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

Console

To delete an Amazon Q data source connector

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In Applications, select the application that you want to delete data sources from.
3. On the application page, from **Data sources**, select the data source that you want to delete.

4. From **Actions**, choose **Delete**.

5. In the dialog box that opens, type **Delete** to confirm deletion, and then choose **Delete**.

   You are returned to the service console while your data source connector is deleted. When the deletion process is complete, the console displays a message confirming successful deletion.

**AWS CLI**

**To delete an Amazon Q data source connector**

```
aws qbusiness delete-data-source \
--application-id application-id \
--index-id index-id \
--data-source-id data-source-id
```

**Getting properties of an Amazon Q data source connector**

To get the properties of an Amazon Q data source connector, you can use the **GetDataSource** API operation.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

**Console**

**To get properties of an Amazon Q data source connector**

1. Sign in to the AWS Management Console and open the Amazon Q console at `https://console.aws.amazon.com/amazonq/`.

2. In **Applications**, select the application you want that contains your data sources.

3. On the application page, from **Data sources**, select the data source that you want to view details for.

4. Under **Data source details**, the following details are available:

   - **Name** – The name of your data source.
• **Status** – The status of your data source.

• **Last sync status** – The status of your last sync.

• **Description** – The description that you gave to your data source.

• **Type** – The type of data source that you're using.

• **Last sync time** – The time that your data source was last synced.

• **Data source ID** – The ID of your data source.

• **IAM role ARN** – The Amazon Resource Name (ARN) of the IAM role that's associated with your data source.

• **Current sync state** – The current sync state of your data source.

**To get Amazon Q data source connector settings**

1. Sign in to the AWS Management Console and open the Amazon Q console at [https://console.aws.amazon.com/amazonq/](https://console.aws.amazon.com/amazonq/).

2. In Applications, select the application you want that contains your data sources.

3. On the application page, from Data sources, select the data source that you want to view details for.

4. For Data source details, choose Settings.

5. For Settings, the following settings are available:

   • **IAM role** – The ARN of the IAM that's associated with your data source.

   • **Sync scope** – The configuration details for your data source.

   • **Sync mode** – The sync type that you chose for your data source.

   • **Sync schedule** – The sync schedule that you chose for your data source.

   • **Field mappings** – The data source document fields that you chose to map to Amazon Q index fields.

**AWS CLI**

**To get Amazon Q data source connector properties**

```bash
aws qbusiness get-data-source \
   --application-id application-id \
```
Listing Amazon Q data source connectors

To list Amazon Q data source connectors, you can use the console or the `ListDataSources` API operation.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

**Console**

**To list Amazon Q data source connectors**

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In Applications, select the application you want that contains your data sources.
3. On the application page, under Data sources, a list of data sources connected to your application is displayed.

**AWS CLI**

**To list Amazon Q data source connectors**

```bash
aws qbusiness list-data-sources \
--application-id application-id \
--index-id index-id \
--max-results maximum-number-of-results-to-return
```

Updating Amazon Q data source connectors

To update your Amazon Q data source connectors, you can use the console or the `UpdateDataSource` API operation.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.
To update a Amazon Q data source connector

Option 1

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In Applications, select the application you want to delete data sources from.
3. On the application page, from Data sources, select the data source that you want to edit.
4. From Actions, choose Edit.
   
   You are redirected to your data source configuration page to edit your existing settings.

Option 2

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In Applications, select the application you want to delete data sources from.
3. On the application page, from Data sources, select the data source that you want to edit.
4. On the data source page, from Actions, choose Edit.
   
   You are redirected to your data source configuration page to edit your existing settings.

CLI

To update your Amazon Q connector

```
aws qbusiness update-data-source \
  --application-id application-id \
  --data-source-id data-source-id \
  --index-id index-id \
  --configuration optional-data-source-configuration-details \
  --description optional-description \
  --display-name optional-display-name \
  --document-enrichment-configuration optional-document-enrichment-configuration \
  --role-arn optional-roleArn \
  --sync-schedule optional-sync-schedule-information \
```
Starting data source connector sync jobs

To start Amazon Q data source connector sync jobs, you can use the console or the `StartDataSourceSyncJobs` API operation.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

Console

**To start your Amazon Q data source connector sync jobs**

1. Sign in to the AWS Management Console and open the Amazon Q console at [https://console.aws.amazon.com/aamzonq/](https://console.aws.amazon.com/aamzonq/).
2. In **Applications**, select the application you want to sync data sources in.
3. On the application page, from **Data sources**, select the data source that you want to sync.
4. Choose **Sync now**.

The console displays a message confirming that your sync job has started successfully.

**Note**

You can also view your sync job report in the Amazon CloudWatch console.

AWS CLI

**To start your Amazon Q data source connector sync jobs**

```bash
aws qbusiness start-data-source-sync-job \
--application-id application-id \
--index-id index-id \
--data-source-id data-source-id
```
Stopping data source connector sync jobs

To stop your Amazon Q connector sync jobs, you can use the console or the \nStopDataSourceSyncJobs API operation.

Note
You can only stop a sync job already in progress.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

Console

To stop your Amazon Q data source connector sync jobs

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In Applications, select the application you want to sync data sources in.
3. On the application page, from Data sources, select the data source that you want to stop the sync for.
4. Choose Stop sync.
5. In the dialog box that opens, type Stop to confirm your action and then select Stop sync.
   The console displays a message confirming that your data source sync job is being stopped.

AWS CLI

To stop your Amazon Q data source connector sync jobs

```bash
aws qbusiness stop-data-source-sync-job
   --application-id application-id
   --data-source-id data-source-id
   --index-id index-id
```
Listing data source connector sync jobs

To list Amazon Q data source connector sync jobs that are in progress, you can use the console or the ListDataSourceSyncJobs API operation.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

Console

To list your Amazon Q data source connector sync jobs

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In Applications, select the application you want that contains your data sources.
3. On the application page, from Data sources, select the data source that you want to view details for.
4. Under Data source details, choose the Sync run history tab.

You will see a list of ongoing, completed, and failed sync jobs for your data sources.

CLI

To list your Amazon Q data source connector sync jobs

```bash
aws qbusiness list-data-source-sync-job \
   --application-id application-id \ 
   --data-source-id data-source-id \ 
   --index-id index-id \ 
   --max-results max-results-to-return
```
Configuring Amazon Q data source connectors

A data source connector is a mechanism for integrating and synchronizing data from multiple repositories into one container index. Amazon Q offers multiple data source connectors that can connect to your data sources and help you create your generative AI solution with minimal configuration.

To configure and connect a data source to your Amazon Q application, use the CreateDataSource API operation. Specify your connector configuration details using the configuration parameter of the CreateDataSource operation. If you use the AWS Management Console instead of the API, you create, configure, and connect your data source as part of the application creation process.

This section contains an overview of data source connector features, recommended best practices for configuration, and configuration information specific to your data source connector.

Topics

- Data source connector concepts
- Best practices for data source connector configuration in Amazon Q
- Supported connectors
- Understanding Amazon Q User Store
- Using Amazon VPC with Amazon Q connectors

Data source connector concepts

This topic outlines specific concepts and features of Amazon Q data source connectors. These concepts are key to understanding how to configure your connector setup. These terms recur on the AWS Management Console, AWS Command Line Interface (AWS CLI), and the Amazon Q API.

Topics

- Source and endpoint metadata
- Authorization
- Authentication
- Virtual private cloud
- Web proxy
• IAM role
• Identity crawler
• Sync scope
• Sync mode
• Sync run schedule
• Field mappings

Source and endpoint metadata

You enter your data source configuration information in the Source section on the console. If you use the API, you specify this information using the configuration parameter of the CreateDataSource operation. Connection configuration information varies depending on the data source. To make sure your connector configures correctly, check the following details:

• You're following connector configuration best practices.
• You've completed the prerequisites for data source configuration. Prerequisites information specific to your data source connector is on each connector's specific page.

Authorization

Amazon Q connectors crawl access control list (ACL) information that's attached to a document along with the document itself. Specifically, the crawled ACL information is the local users and local groups that have access to the document.

Then, Amazon Q uses the ACL information it crawls, together with the Identity crawler feature, to generate chat responses for your end user. The responses are based on the documents that the user has access to. For more information about this process, see Understanding User Store.

During connector configuration, you can choose either to activate or deactivate the Authorization feature. If you activate this feature, Amazon Q will only generate content for an end user from documents they have access to. If this feature is deactivated, the connector won't crawl any ACL information and all documents will be considered public to all application users.

If you choose to crawl an ACL, the connector updates any changes in ACLs each time that your data source content is crawled. To capture ACL changes to make sure that the right end users have access to the right content, re-sync your data source regularly.
Authentication

To authenticate Amazon Q to access your data source, you provide your data source access credentials to Amazon Q using an AWS Secrets Manager secret. If you use the console, you can choose to create a new secret or use an existing one. If you use the API, you must provide the Amazon Resource Name (ARN) of an existing Secrets Manager secret when you use the CreateDataSource operation.

Note
You should regularly refresh or rotate your credentials and secret details. Provide only the necessary access level for your own security. Don't re-use credentials and secrets across data sources.

For on-premises or server data source connectors, Amazon Q checks if the endpoint information included in Secrets Manager is the same endpoint information specified in your data source configuration details. This helps protect against the confused deputy problem, which is a security issue. The problem occurs when a user doesn't have permissions to perform an action. But, by using Amazon Q as a proxy, the user can access the configured secret and perform the action.

If you change your endpoint information later, you must create a new secret to sync this information.

Note
If you change your authentication type and credentials, you must update your IAM role to access the correct Secrets Manager secret ID.

Virtual private cloud

Amazon Q can connect to Amazon Virtual Private Cloud to index content stored in data sources or databases running in your private cloud. If your data source or database isn't running on Amazon VPC, you can connect your data source or database to Amazon VPC using a virtual private network (VPN).

You can use Amazon VPC with either the console or the Amazon Q API. If you're using the API, you specify the vpcConfiguration when you use the CreateDataSource API operation.
If you're using Amazon VPC with Amazon Q, you need the following information:

- The identifier of the subnet that contains the data source.
- The identifier of the security groups that grant access to the host.
- An IAM role with access to Amazon VPC and permissions to create and delete an elastic network interface in your subnets is also required.

You can find the subnet and security group IDs in the Amazon VPC console. For more information, see [What is Amazon VPC?](https://docs.aws.amazon.com/vpc/latest/userguide) in the *Amazon VPC User Guide*.

For more information about using Amazon VPC with Amazon Q, see [Using Amazon VPC with connectors.](https://docs.aws.amazon.com/q/latest/devguide/using-amazon-vpc.html).

### Web proxy

For all supported data sources, you can use a web proxy to connect to your data source instance. You must provide the host name and port number. For example, `a.example.com` is the hostname of `https://a.example.com/page1.html`, and the port is 443, which is the standard port for HTTPS.

⚠️ **Important**

For security reasons, Amazon Q only supports web proxy using HTTPS protocol.

### IAM role

To create your data source connector, Amazon Q requires permissions to interact with other services.

If you're using the console, you can choose an existing IAM role or let Amazon Q create a role for you. If you're unsure if an existing role is used for an application, choose **Create a new role** to avoid an error.

⚠️ **Note**

To **Create a new role** during connector configuration on the console, you must have permissions to create an IAM role.
If you're using the API, you must provide the ARN of an existing IAM role when you use the **CreateDataSource** operation.

IAM roles used for applications can't be used for data sources.

### Note

Make sure your IAM role includes the permissions to support your Amazon Q connector configurations.

---

## Identity crawler

Amazon Q crawls ACL information at the document level from supported data sources (users and group access to documents). In addition, Amazon Q crawls the local user and group identity configurations within each data source. This approach is useful where your application is connected to multiple data sources with different authorization and authentication systems, but you want to create a unified, access-controlled chat experience for end users.

Amazon Q stores the principal information in the Amazon Q user store. Then, Amazon Q internally maps the local user and group IDs attached to the document to the federated identities of users and groups. Mapping identities streamlines user management and speeds up chat responses by reducing ACL information retrieval time during chat requests.

Identity crawling, along with the **Authorization** feature, helps to filter and generate web experience content restricted by end user context. For more information about this process, see [Understanding User Store](#).

### Note

To activate the **Identity crawler** feature on the console, you must also activate ACL crawling as part of the **Authorization** feature. If you use the API, you choose to activate identity crawling using the configuration object when you use the **CreateDataSource** operation.
Sync scope

You can choose to customize the content crawled and indexed by your data source connector. The sync scope options available vary based on the data source connector.

Sync mode

With sync mode, you can customize what content gets synced with your index when your data source content changes. Choose from the following options:

**Console**

- **Full sync** – Sync all content regardless of the previous sync status.
- **New or modified content sync** – Sync only new or modified documents.
- **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.
- **Change log** – Crawl and sync only new, modified, and deleted content.

**API**

Specify the sync mode using the `configuration` parameter of the `CreateDataSource` operation. Choose from the following options:

- **Forced full crawl** – Crawl and sync all content to your index.
- **Full crawl** – Crawl all content and sync only new, modified, or deleted content.
- **Change log** – Crawl and sync only new, modified, and deleted content.

**Note**

Available sync mode features vary across data source connectors.

**Important**

If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it's because the CloudWatch logs aren't available yet. Wait for some time and check again.
Sync run schedule

When you use the console or the CreateDataSource API operation, you can choose to periodically sync your data source with your retriever on a custom schedule. You can choose from the following frequency options:

- **Run on demand** – Sync a data source with your index only when you choose to.
- **Hourly** – Sync your data source with your index every hour. You can choose which minute the sync begins.
- **Daily** – Sync your data source with your index daily. You can choose the sync start time in UTC format in hours and minutes.
- **Weekly** – Sync your data source with your index weekly. You can choose the days to sync and the sync start time in hours and minutes (UTC format).
- **Monthly** – Sync your data source monthly with your index. You can choose the day of the month to start the sync and the sync start time in hours and minutes (UTC format).
- **Custom** – Sync your data source to your index using a cron expression. A cron expression is a string comprising five or six required fields, separated by white space. Cron expressions represent a set of times programmed to schedule events. For example, an expression to activate a rule every day at 12:00pm UTC can look like: (0 12 * * ? *). Similarly, an expression to activate a rule every day at 10:15am UTC on the last Friday of each month during the years 2023 to 2025 can look like: (15 10 ? * 6L 2023-2025).

**Note**

Amazon Q will not sync the data source (even for the first time) until you select **Sync now** after you successfully add the data source.

Field mappings

When you connect Amazon Q to your data, your data source connector crawls relevant metadata or attributes associated with a document. Examples of metadata include date of creation, document id, and document name. Then, Amazon Q maps the metadata to fields within your Amazon Q index.
You map data source document attributes to Amazon Q index fields using the Field mappings feature on the console, or the configuration parameter of the CreateDataSource API operation. If you use the console, you add field mappings after your data source is created.

For more information, see the following topics:

- Document attributes and types
- Filtering using metadata

### Best practices for data source connector configuration in Amazon Q

The following list describes best practices for setting up and configuring your Amazon Q data source connector:

- Each document in an index must be unique. Check that there are no duplicate documents in a data source, or across any data sources, that you plan to connect to an Amazon Q retriever.
- If you change your authentication type and credentials, you must update your IAM role to access the correct AWS Secrets Manager secret ID.
- We recommend that you regularly refresh or rotate your credentials and secret. Provide only the necessary access level for your own security. We don’t recommend the re-use of credentials and secrets across data sources.
- IAM roles used for retrievers can’t be used for data sources. If you’re unsure if an existing role is used for a retriever or data source, create a new IAM role to avoid errors.
- If you use AWS KMS keys for the application, ensure that the IAM for your application is given the permission to describe, encrypt, and decrypt data using this key.
- For on-premises or server data source connectors, Amazon Q checks if the endpoint information included in Secrets Manager is the same as the endpoint information specified in your data source configuration details. This helps protect against the confused deputy problem, which is a security issue. The problem occurs when a user doesn’t have permission to perform an action. But, by using Amazon Q as a proxy, the user can access the configured secret and perform the action.

If you change your endpoint information later, you must create a new secret to sync this information.
• 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. If you specify an inclusion and exclusion filter, documents that match the exclusion filter aren't indexed, even if they match the inclusion filter.

**Supported connectors**

Amazon Q currently supports the following connectors:

• [AEM (Cloud)](https://aws.amazon.com/q/q-connectors/aem-cloud/)
• [AEM (Server)](https://aws.amazon.com/q/q-connectors/aem-server/)
• [Alfresco (Cloud)](https://aws.amazon.com/q/q-connectors/alfresco-cloud/)
• [Alfresco (Server)](https://aws.amazon.com/q/q-connectors/alfresco-server/)
• [Aurora (MySQL)](https://aws.amazon.com/q/q-connectors/aurora-mysql/)
• [Aurora (PostgreSQL)](https://aws.amazon.com/q/q-connectors/aurora-postgresql/)
• [Amazon FSx Windows](https://aws.amazon.com/q/q-connectors/amazon-fsx-windows/)
• [Amazon RDS (Microsoft SQL Server)](https://aws.amazon.com/q/q-connectors/amazon-rds-mssql/)
• [Amazon RDS (MySQL)](https://aws.amazon.com/q/q-connectors/amazon-rds-mysql/)
• [Amazon RDS (Oracle)](https://aws.amazon.com/q/q-connectors/amazon-rds-oracle/)
• [Amazon RDS (PostgreSQL)](https://aws.amazon.com/q/q-connectors/amazon-rds-postgresql/)
• [Amazon S3](https://aws.amazon.com/q/q-connectors/amazon-s3/)
• [Amazon Q custom connector](https://aws.amazon.com/q/q-connectors/amazon-q-custom/)
• [Amazon Q Web Crawler](https://aws.amazon.com/q/q-connectors/amazon-q-web-crawler/)
• [Amazon WorkDocs](https://aws.amazon.com/q/q-connectors/amazon-workdocs/)
• [Box](https://aws.amazon.com/q/q-connectors/box/)
• [Confluence (Cloud)](https://aws.amazon.com/q/q-connectors/confluence-cloud/)
• [Confluence (Server)](https://aws.amazon.com/q/q-connectors/confluence-server/)
• [Dropbox](https://aws.amazon.com/q/q-connectors/dropbox/)
• [Drupal](https://aws.amazon.com/q/q-connectors/drupal/)
• [GitHub (Cloud)](https://aws.amazon.com/q/q-connectors/github-cloud/)
• [Drupal](https://aws.amazon.com/q/q-connectors/drupal/)
Connecting AEM (Cloud) to Amazon Q

Adobe Experience Manager (AEM) is a content management system (CMS) that’s used for creating website or mobile app content. You can connect your AEM Cloud instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
• For information about connector configuration best practices, see Connector configuration best practices.

Topics
• Prerequisites
• Using the console
• Using the API
• ACL crawling
• IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In AEM, make sure you have:

• Access to an account with administrative permissions, or are an admin user.
• Copied your AEM Cloud host URL.
• Noted your basic authentication credentials of admin username and password.
• Optional: Generated OAuth 2.0 credentials in AEM Cloud. 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 Cloud, see AEM Cloud documentation.

In your AWS account, make sure you have:

• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
• Stored your AEM Cloud authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.
For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Adobe Experience Manager page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   **Note**: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Source** – Choose AEM as a Cloud Service.
   
   - **AEM host URL** – Enter your AEM host URL. If you use AEM as a Cloud Service, you can use the author URL. For example: https://author-xxxxx-xxxxxx-adobeaemcloud.com.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.

   **Note**: Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).

4. **Authentication** – Choose between Basic authentication and OAuth 2.0 authentication and then enter the following information for your AWS Secrets Manager secret.
   
   a. **Basic authentication** – Enter a name for the secret, your AEM site admin username, and admin password.
   
   b. **OAuth 2.0 authentication** – Enter a name for the secret, your client ID, client secret, private key, organization ID, technical account ID, and Adobe IMS host.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:
   
   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

For more information, see [VPC](#).

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

For more information, see [IAM role](#).

8. In **Sync scope**, enter the following information:

   a. **Sync content types** – Choose whether to crawl only **Pages** or **Assets**, or both.

   b. **Additional configuration – optional** – 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 AEM template editor and allows page header and 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 AEM. They allow you to prepare content ready for use in multiple locations and 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.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   • **Full sync** – Sync all content regardless of the previous sync status.

   • **New or modified content sync** – Sync only new and modified documents.

   • **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.
For more details, see **Sync mode**.

10. **In Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see **Sync run schedule**.

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See **Tags** for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see **Field mappings**.

13. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

   **Note**

   You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the **CreateDataSource** action to connect a data source to your Amazon Q application.
Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**AEM JSON schema**

The following is the AEM JSON schema:

```json
{
    "$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": [
                "aemUrl",
                "authType",
                "deploymentType"
            ]
        }
    },
    "required": [
        "connectionConfiguration"
    ]
}
```
"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",
                "dateFieldFormat"
              ]
            }
          ]
        }
      }
    }
  }
},
"AEM (Cloud)"
"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": {
    "isCrawlAcl": {
      "type": "boolean"
    },
    "fieldForUserId": {
      "type": "string"
    },
    "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/Lusaka",
        "Africa/Mbabane",
        "Africa/Malabo",
        "Africa/Monrovia",
        "Africa/Nairobi",
        "Africa/Niamey",
        "Africa/Nouakchott",
        "Africa/Nyanya",
        "Africa/Ouagadougou",
        "Africa/Pretoria",
        "Africa/Riyadh",
        "Africa/Sao_Tome",
        "Africa/Tripoli",
        "Africa/Tunis",
        "Africa/Tripoli",
        "Africa/Windhoek",
        "Africa/Zambesia",
        "America/Anchorage",
        "America/Argentina/Buenos_Aires",
        "America/Argentina/Cordoba",
        "America/Argentina/Jujuy",
        "America/Argentina/Mendoza",
        "America/Argentina/Rio_Qu�en",
        "America/Argentina/Salta",
        "America/Argentina/San_Luis",
        "America/Argentina/San_Rafael",
        "America/Argentina/Tucuman",
        "America/Argentina/Ushuaia",
        "America/Aruba",
        "America/Asuncion",
        "America/Bahia",
        "America/Bahia_Beauchene",
        "America/Barbados",
        "America/Belem",
        "America/ Bipab",
        "America/Blumenau",
        "America/Boa_Vista",
        "America/Brasilia",
        "America/Brasilia",
        "America/Buenos Aires",
        "America/Ceara",
        "America/Chauncy",
        "America/Chihuahua",
        "America/Ciudad_Rica",
        "America/Cocibolca",
        "America/Cozumel",
        "America/ Eagle_Woods",
        "America/El_Salvador",
        "America/Ensenada",
        "America/Fortaleza",
        "America/Guatemala",
        "America/Guatemala",
        "America/Guyanas",
        "America/ Iquitos",
        "America/Jamaica",
        "America/Juneau",
        "America/Kralendijk",
        "America/La_Paz",
        "America/Lima",
        "America/Los_Angeles",
        "America/Lucama",
        "America/La_Paz",
        "America/Lusaka",
        "America/Menominee",
        "America/Merida",
        "America/Montevideo",
        "America/Monterrey",
        "America/Monterrey",
        "America/Montevideo",
        "America/Morristown",
        "America/Namibia",
        "America/Nassau",
        "America/Negros",
        "America/New_York",
        "America/Niagara_Falls",
        "America/Nicaragua",
        "America/Nipigon",
        "America/Norman",
        "America/Nunavut",
        "America/ OchoaNieves",
        "America/Oranjestad",
        "America/Panama",
        "America/Paramaribo",
        "America/Phoenix",
        "America/Porto_Alegre",
        "America/ Portoviejo",
        "America/Quito",
        "America/Rainforest",
        "America/Rankin_Inlet",
        "America/Rio_Branco",
        "America/Rosario",
        "America/Thule",
        "America/Tegucigalpa",
        "America/Tijuana",
        "America/Toronto",
        "America/Torres",
        "America/Ubangi",
        "America/Unica",
        "America/Union",
        "America/ Usulutan",
        "America/Vancouver",
        "America/Veracruz",
        "America/ Willemstad",
        "America/Winnipeg",
        "America/Withney",
        "America/Yakutat",
        "America/Yellowknife",
        "America/Yucatan",
        "America/Yupik",
        "America/Zacatecas",
        "America/Zamora"
      ]
    }
  }
}
"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","AEM (Cloud)
"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_Prines",
"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/Resolute",
"America/Recife",
"America/Regina",
"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/Kandyga",
"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/P'yongyang",
"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", 

AEM (Cloud)
"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/Canberra",
"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",
"GB"
"Libya",
"MET",
"MST7MDT",
"Mexico/BajaNorte",
"Mexico/BajaSur",
"Mexico/General",
"NZ",
"NZ-CHAT",
"Navajo",
"PRC",
"PST8PDT",
"Pacific/Api",
"Pacific/Auckland",
"Pacific/Bougainville",
"Pacific/Chatham",
"Pacific/Chuuk",
"Pacific/Easter",
"Pacific/Eden",
"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",
"AEM (Cloud)
"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"
},
  "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"
    }
  }}
"type": "array",
"items": {
  "type": "string"
}
],
"resourceExclusionPatterns": {
  "type": "array",
  "items": {
    "type": "string"
  }
},
"resourceInclusionPatterns": {
  "type": "array",
  "items": {
    "type": "string"
  }
},
"resourceExclusionPatterns": {
  "type": "array",
  "items": {
    "type": "string"
  }
},
"pageComponents": {
  "type": "array",
  "items": {
    "type": "object"
  }
},
"contentFragmentVariations": {
  "type": "array",
  "items": {
    "type": "object"
  }
},
"cugExemptedPrincipals": {
  "type": "array",
  "items": {
    "type": "string"
  }
},
"required": []}
The following table provides information about important JSON keys to configure.
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>aemUrl</td>
<td>The Adobe Experience Manager host URL. For example, if you use AEM On-Premise, you include the hostname and port: <a href="http://hostname:port">http://hostname:port</a>. Or, if you use AEM as a Cloud Service, you can use the author URL: <a href="https://author-xxxxx-xxxxxxx.adobeaemcloud.com">https://author-xxxxx-xxxxxxx.adobeaemcloud.com</a>.</td>
</tr>
<tr>
<td>authType</td>
<td>The type of authentication you use, whether Basic or OAuth2.</td>
</tr>
<tr>
<td>deploymentType</td>
<td>The type of Adobe Experience Manager that you use, either CLOUD or ON-PREMISE.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• page</td>
<td>A list of objects that map the attributes or field names of your Adobe Experience Manager pages and assets to Amazon Q index field names.</td>
</tr>
<tr>
<td>• asset</td>
<td></td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>timeZoneId</td>
<td>If you use AEM On-Premise and the time zone of your server is different than the time zone of the Amazon Q 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 Q AEM connector or index. The default time zone for AEM as a Cloud Service is Greenwich Mean Time.</td>
</tr>
<tr>
<td>pageRootPaths, assetRootPaths</td>
<td>A list of root paths for pages and assets. For example, the root path for a page could be <code>/content/sub</code> and the root path for an asset could be <code>/content/sub/asset1</code>.</td>
</tr>
<tr>
<td>crawlAssets</td>
<td>Specify true to crawl assets.</td>
</tr>
<tr>
<td>crawlPages</td>
<td>Specify true to crawl pages.</td>
</tr>
<tr>
<td>pagePathInclusionPatterns, pageNameInclusionPatterns, assetPathInclusionPatterns, assetTypeInclusionPatterns, assetNameInclusionPatterns</td>
<td>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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• pagePathExclusionPatterns</td>
<td>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.</td>
</tr>
<tr>
<td>• pageNameExclusionPatterns</td>
<td>A list of names for the specific page components that you want to index.</td>
</tr>
<tr>
<td>• assetPathExclusionPatterns</td>
<td>A list of names for the specific saved variations of Adobe Experience Manager Content Fragments that you want to index.</td>
</tr>
<tr>
<td>• assetTypeInclusionPatterns</td>
<td>The type of data source. Specify AEM as your data source type.</td>
</tr>
<tr>
<td>• assetNameInclusionPatterns</td>
<td>Specify true to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to specific documents. See Identity crawler for more information.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>• Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</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 Adobe Experience Manager. The secret must contain a JSON structure with the following keys:</td>
</tr>
</tbody>
</table>

If using basic authentication for either AEM On-Premise or Cloud:

```json
{
    "aemUrl": "Adobe Experience Manager On-Premise host URL ",
    "username": " user name with admin permissions ",
    "password": " password with admin permissions "
}
```

If using OAuth 2.0 authentication for AEM On-Premise:

```json
{
    "aemUrl": "Adobe Experience Manager host URL",
    "clientId": " client ID",
    "clientSecret": " client secret",
    "privateKey": " private key"
}
```

If using OAuth 2.0 authentication for AEM as a Cloud Service:

```json
{
    "clientId": " client ID",
    "clientSecret": " client secret",
    "privateKey": " private key",
    "orgId": "organization ID ",
```
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>technical Account ID</td>
<td>&quot;technical Account ID&quot;</td>
</tr>
<tr>
<td>IMS Host</td>
<td>&quot;Adobe Identity Management System (IMS) host&quot;</td>
</tr>
</tbody>
</table>

| version | The version of this template that's currently supported. |

### ACL crawling

When you connect an AEM data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your AEM instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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're mapped from the names of the groups in AEM.
- **_user_id** – User IDs exist in Adobe Experience Manager content where there are set access permissions. They're mapped from the user emails as the IDs in AEM.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

### IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
• Permission to access the User Store API operations to ingest user and group access control information from documents.

• Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.

• (Optional) If you're using Amazon VPC, permission to access your Amazon VPC.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowsAmazonQToGetSecret",
         "Effect": "Allow",
         "Action": [
            "secretsmanager:GetSecretValue"
         ],
         "Resource": [
            "arn:aws:secretsmanager:{region}:{account_id}:secret:[[secret_id]]"
         ]
      },
      {
         "Sid": "AllowsAmazonQToDecryptSecret",
         "Effect": "Allow",
         "Action": [
            "kms:Decrypt"
         ],
         "Resource": [
            "arn:aws:kms:{region}:{account_id}:key:[[key_id]]"
         ],
         "Condition": {
            "StringLike": {
               "kms:ViaService": [
                  "secretsmanager.*.amazonaws.com"
               ]
            }
         }
      },
      {
         "Sid": "AllowsAmazonQToIngestDocuments",
         "Effect": "Allow",
         "Action": [
            "qbusiness:BatchPutDocument",
```
"qbusiness:BatchDeleteDocument"
],
"Resource": "arn:aws:qbusiness:{region}:{{source_account}}:application/{application_id}/index/{{index_id}}"
},
{
"Sid": "AllowsAmazonQToIngestPrincipalMapping",
"Effect": "Allow",
"Action": [
"qbusiness:PutGroup",
"qbusiness:CreateUser",
"qbusiness:DeleteGroup",
"qbusiness:UpdateUser",
"qbusiness:ListGroup"
],
"Resource": [
"arn:aws:qbusiness:{region}:{{account_id}}:application/{{application_id}}",
"arn:aws:qbusiness:{region}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
"arn:aws:qbusiness:{region}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/*"
]
},
{
"Sid": "AllowsAmazonQToCreateAndDeleteNI",
"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]]"
]
},
{
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterface",
"ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{region}:{{account_id}}:network-interface/*",
"Condition": {
"StringLike": {
"aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
},
"ForAllValues:StringEquals": {
"aws:TagKeys": [
"AMAZON_Q"
]
}
},
{
"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
"ec2:CreateTags"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringEquals": {
"ec2:CreateAction": "CreateNetworkInterface"
}
}
},
{
"Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterfacePermission"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringLike": {
"aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
}
}
},
{
"Sid": "AllowsAmazonQToDescribeResourcesForVPC",
"Effect": "Allow",
"Action": [
"ec2:DescribeNetworkInterfaces",
"ec2:DescribeAvailabilityZones",
"ec2:DescribeNetworkInterfaceAttribute",
"ec2:DescribeVpcs",
"ec2:DescribeTags"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringLike": {
"aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
}
}
},
{
"Sid": "AllowsAmazonQToDescribeTags",
"Effect": "Allow",
"Action": [
"ec2:DescribeTags"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringLike": {
"aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
}
}
}
"ec2:DescribeRegions",
"ec2:DescribeNetworkInterfacePermissions",
"ec2:DescribeSubnets"
],
"Resource": "*
}
]
}

To allow Amazon Q to assume a role, you must also use the following trust policy:

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:
{{source_account}}:application/{{application_id}}"
        }
      }
    }
  ]
}

For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting AEM On-Premise to Amazon Q

Adobe Experience Manager (AEM) is a content management system (CMS) that's used for creating website or mobile app content. You can connect AEM On-Premise instance to Amazon Q—using
either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In AEM, make sure you have:

- Access to an account with administrative permissions, or an admin user.
- Copied your AEM host URL.
- Noted your basic authentication credentials of admin username and password.
- **Optional**: Generated OAuth 2.0 credentials in AEM On-Premise. If you use AEM On-Premise, the credentials include client ID, client secret, and private key. Adobe Granite OAuth 2.0 server implementation (com.adobe.granite.oauth.server) provides the support for OAuth 2.0 server functionalities in AEM.

In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
• Stored your AEM On-Premise authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Adobe Experience Manager page, enter the following information:

1. Name – Name your data source for easy tracking.

   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. Source – Choose AEM On-Premise.

   a. AEM host URL – Enter your AEM host URL. If you use AEM On-Premise, you include the hostname and port. For example: https://hostname:port.

   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.

3. Authorization – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.

   Note
Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).

4. Authentication – Choose between Basic authentication and OAuth 2.0 authentication and then enter the following information for your AWS Secrets Manager secret.
a. **Basic authentication** – Enter the name for your secret, your AEM site admin username, and admin password.

b. **OAuth 2.0 authentication** – Enter enter a name for the secret, your client ID, client secret, and private key.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

8. **In Sync scope**, enter the following information:

   a. **Sync content types** – Choose whether to crawl only Pages or Assets, or both.

   b. **Additional configuration – optional** – 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 AEM template editor and allows page header and 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 AEM. They allow you to prepare content ready for use in multiple locations and 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.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New or modified content sync** – Sync only new and modified documents.
   - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

   For more details, see [Sync mode](#).

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the [Data source details](#) page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.

    You are taken to the [Data source details](#), where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.
You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

## Using the API

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the `configuration` parameter to provide a JSON schema with all other configuration information specific to your data source connector.

### AEM JSON schema

The following is the AEM JSON schema:

```json
{
    "$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"
                        }
                    }
                }
            }
        }
    }
}
```
"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"
                }
              }
            }
          ]
        },
        "dataSourceFieldId": {
          "type": "string"
        }
      }
    }
  }
}
"dateFieldFormat": {
  "type": "string",
  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
},
"required": [
  "indexFieldName",
  "indexFieldType",
  "dataSourceFieldName"
],
"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"
            }
          }
        }
      ]
    }
  }
}
"additionalProperties": {
  "type": "object",
  "properties": {
    "isCrawlAcl": {
      "type": "boolean"
    },
    "fieldForUserId": {
      "type": "string"
    },
    "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/Conakry",
        "Africa/Douala",
        "Africa/East_Africa",
        "Africa/Freetown",
        "Africa/Gaborone",
        "Africa/Georgetown",
        "Africa/Johannesburg",
        "Africa/Kinshasa",
        "Africa/Lagos",
        "Africa/Libreville",
        "Africa/Lome",
        "Africa/Nairobi",
        "Africa/Nouakchott",
        "Africa/Rome",
        "Africa/Sao_Tome_and_Principe",
        "Africa/Tripoli",
        "Africa/Tunis",
        "Africa/Tripoli",
        "Africa/Windhoek"
      ]
    }
  }
}
"Africa/Cairo",
"Africa/Casablanca",
"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/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/Minominue",
"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/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",

AEM (Server) 137
"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/Kwajalein",
"Pacific/Midway",
"Pacific/Nauru",
"Pacific/PortMoresby",
"Pacific/Rarotonga",
"Pacific/StHelen",
"Pacific/Tongatapu",
"Pacific/Chuuk",
"Pacific/Chuuk/
"Pacific/Fiji",
"Pacific/Honolulu",
“Pacific/Kantol",
“Pacific/Midway",
“Pacific/Nauru",
“Pacific/PortMoresby",
“Pacific/Rarotonga",
“Pacific/StHelen",
“Pacific/Tongatapu",
“Pacific/Chuuk",
“Pacific/Chuuk/
"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"
    }
  ]
},
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>aemUrl</td>
<td>The Adobe Experience Manager host URL. For example, if you use AEM On-Premise, you include the hostname and port: <a href="http://hostname:port">http://hostname:port</a>. Or, if you use AEM as a Cloud Service, you can use the author URL: <a href="https://author-xxxxxx-xxxxxxx.adobeaemcloud.com">https://author-xxxxxx-xxxxxxx.adobeaemcloud.com</a>.</td>
</tr>
<tr>
<td>authType</td>
<td>The type of authentication you use, whether Basic or OAuth2.</td>
</tr>
<tr>
<td>deploymentType</td>
<td>The type of Adobe Experience Manager that you use, either CLOUD or ON-PREMISE.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>page</td>
<td>A list of objects that map the attributes or field names of your Adobe Experience Manager pages and assets to Amazon Q index field names.</td>
</tr>
<tr>
<td>asset</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>timeZoneId</td>
<td>If you use AEM On-Premise and the time zone of your server is different than the time zone of the Amazon Q 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 Q AEM connector or index. The default time zone for AEM as a Cloud Service is Greenwich Mean Time.</td>
</tr>
<tr>
<td>pageRootPaths</td>
<td>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.</td>
</tr>
<tr>
<td>assetRootPaths</td>
<td></td>
</tr>
<tr>
<td>crawlAssets</td>
<td>Specify true to crawl assets.</td>
</tr>
<tr>
<td>crawlPages</td>
<td>Specify true to crawl pages.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• pagePathInclusionPatterns</td>
<td>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.</td>
</tr>
<tr>
<td>• pageNameInclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• assetPathInclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• assetTypeInclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• assetNameInclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• pagePathExclusionPatterns</td>
<td>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.</td>
</tr>
<tr>
<td>• pageNameExclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• assetPathExclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• assetTypeInclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• assetNameInclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>pageComponents</td>
<td>A list of names for the specific page components that you want to index.</td>
</tr>
<tr>
<td>contentFragmentVariations</td>
<td>A list of names for the specific saved variations of Adobe Experience Manager Content Fragments that you want to index.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify AEM as your data source type.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>Specify <code>true</code> to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to specific documents. See Identity crawler for more information.</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>• Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</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 Adobe Experience Manager. The secret must contain a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>If using basic authentication for either AEM On-Premise or Cloud:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;aemUrl&quot;: &quot;Adobe Experience Manager On-Premise host URL &quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;username&quot;: &quot; user name with admin permissions &quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;password&quot;: &quot; password with admin permissions &quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>If using OAuth 2.0 authentication for AEM On-Premise:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;aemUrl&quot;: &quot;Adobe Experience Manager host URL&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;clientId&quot;: &quot; client ID&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;clientSecret&quot;: &quot; client secret&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;privateKey&quot;: &quot; private key&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>If using OAuth 2.0 authentication for AEM as a Cloud Service:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;clientId&quot;: &quot; client ID&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;clientSecret&quot;: &quot; client secret&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;privateKey&quot;: &quot; private key&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;orgId&quot;: &quot;organization ID &quot;,</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;technicalAccountId&quot;: &quot;technical account ID&quot;, &quot;imsHost&quot;: &quot;Adobe Identity Management System (IMS) host &quot;</td>
<td></td>
</tr>
</tbody>
</table>

**version**

The version of this template that's currently supported.

### ACL crawling

When you connect an AEM data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your AEM instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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're mapped from the names of the groups in AEM.
- **_user_id** – User IDs exist in Adobe Experience Manager content where there are set access permissions. They're mapped from the user emails as the IDs in AEM.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

### IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.

- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.

- Permission to access the SSL certificate stored in your Amazon S3 bucket.

- (Optional) If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
   "Version": "2012-10-17",
   "Statement": [{
      "Sid": "AllowsAmazonQToGetS3Objects",
      "Action": [ "s3:GetObject" ],
      "Resource": [ "arn:aws:s3:::{{input_bucket_name}}/*" ],
      "Effect": "Allow",
      "Condition": {
         "StringEquals": {
            "aws:ResourceAccount": "{{account_id}}"
         }
      }
   },
   {
      "Sid": "AllowsAmazonQToGetSecret",
      "Effect": "Allow",
      "Action": [ "secretsmanager:GetSecretValue" ],
      "Resource": [ "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]" ]
   },
   {
      "Sid": "AllowsAmazonQToDecryptSecret",
      "Effect": "Allow",
      "Action": [ "kms:Decrypt" ],
      "Resource": [ 
```
"arn:aws:kms:{{region}}:{{account_id}}:key/[[key_id]]",
"Condition": {
  "StringLike": {
    "kms:ViaService": [
      "secretsmanager.*.amazonaws.com"
    ]
  }
},

"Sid": "AllowsAmazonQToIngestDocuments",
"Effect": "Allow",
"Action": [
  "qbusiness:BatchPutDocument",
  "qbusiness:BatchDeleteDocument"
],
"Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},

"Sid": "AllowsAmazonQToIngestPrincipalMapping",
"Effect": "Allow",
"Action": [
  "qbusiness:PutGroup",
  "qbusiness:CreateUser",
  "qbusiness:DeleteGroup",
  "qbusiness:UpdateUser",
  "qbusiness:ListGroups"
],
"Resource": [
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**"
]
},

"Sid": "AllowsAmazonQToCreateAndDeleteNI",
"Effect": "Allow",
"Action": [
  "ec2:CreateNetworkInterface",
  "ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*"
"ec2:DeleteNetworkInterface"
],
"Resource": [
  "arn:aws:ec2:{region}:{account_id}:subnet/[[subnet_ids]]",
  "arn:aws:ec2:{region}:{account_id}:security-group/[[security_group]]"
],
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:{region}:{account_id}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:RequestTag/AMAZON_Q": "qbusiness_{account_id}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "AMAZON_Q"
      ]
    }
  }
}

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

{
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:{region}:{account_id}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:RequestTag/AMAZON_Q": "qbusiness_{account_id}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "AMAZON_Q"
      ]
    }
  }
}
"Action": [
  "ec2:CreateNetworkInterfacePermission"
],
"Resource": "arn:aws:ec2:{region}:{{account_id}}:network-interface/*",
"Condition": {
  "StringLike": {
    "aws:ResourceTag/AMAZON_Q":
    "qbusiness_{{account_id}}_{{application_id}}_*"
  }
}
},
{
  "Sid": " AllowsAmazonQToDescribeResourcesForVPC",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeNetworkInterfaceAttribute",
    "ec2:DescribeVpcs",
    "ec2:DescribeRegions",
    "ec2:DescribeNetworkInterfacePermissions",
    "ec2:DescribeSubnets"
  ],
  "Resource": "*"
}
]

To allow Amazon Q to assume a role, you must also use the following trust policy:

{  
  "Version": "2012-10-17",
  "Statement": [
    
    
    "Sid": " AllowsAmazonQToAssumeRoleForServicePrincipal",
    "Effect": "Allow",
    "Principal": {
      "Service": "qbusiness.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "{{source_account}}"
      }
    }
  ]
}
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting Alfresco (Cloud) to Amazon Q

Alfresco is a content management service (CMS) that helps customers store and manage their content. You can connect Alfresco Cloud instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.
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 username 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.
- **Optional:** Generated OAuth 2.0 credentials in Alfresco. The credentials include client ID, client secret, and token URL. For more information about 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.

In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your Alfresco Cloud authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

**Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

Using the console

On the Alfresco page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Source** – Choose Alfresco Cloud.
a. **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`.

b. **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`.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   ![Note](image)

   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

4. **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 username, and password.

   If you chose **OAuth 2.0 authentication**, enter a name for the secret, client ID, client secret, and token URL.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).
7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

8. In **Sync scope**, enter the following information:
   
   a. **Content** – Choose whether to crawl content marked with 'Aspects' in Alfresco, content within a specific Alfresco site, or content across all your Alfresco sites.
   
   b. **Additional configuration – optional** – 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.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.
You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

**Note**

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the **CreateDataSource** action to connect a data source to your Amazon Q application.

Then, you use the **configuration** parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**Alfresco JSON schema**

The following is the Alfresco JSON schema:

```json
{
   "$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"
      ]
    }
  }
},
"repositoryConfigurations": {
  "type": "object",
  "properties": {
    "document": {
      "type": "object",
      "properties": {
        "Alfresco (Cloud)": 164
      }
    }
  }
},
"required": ["repositoryEndpointMetadata"]
"repositoryConfigurations": {
  "type": "object",
  "properties": {
    "document": {
      "type": "object",
      "properties": {
        "Alfresco (Cloud)": 164
      }
    }
  }
},
"required": ["repositoryEndpointMetadata"]
"repositoryConfigurations": {
  "type": "object",
  "properties": {
    "document": {
      "type": "object",
      "properties": {
        "Alfresco (Cloud)": 164
      }
    }
  }
},
"required": ["repositoryEndpointMetadata"]
"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"
                    },
                    "dateFormatFormat": {
                        "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"
  },
  "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"
]}

The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>siteId</td>
<td>The identifier of the Alfresco site.</td>
</tr>
<tr>
<td>repoUrl</td>
<td>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 <a href="https://company.alfrescocloud.com">https://company.alfrescocloud.com</a>. Or, if you use Alfresco On-Premises, the repository URL could be <a href="https://company-alfresco-instance.company-domain.suffix:port">https://company-alfresco-instance.company-domain.suffix:port</a>.</td>
</tr>
<tr>
<td>webAppUrl</td>
<td>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 <a href="https://example.com">https://example.com</a>.</td>
</tr>
<tr>
<td>repositoryAdditionalProperties</td>
<td>Additional properties for content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>authType</td>
<td>The type of authentication that you use, whether OAuth2 or Basic.</td>
</tr>
<tr>
<td>type (deployment)</td>
<td>The type of Alfresco that you use, whether PAAS or ON-PREM.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>crawlType</td>
<td>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).</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• document</td>
<td>A list of objects that map the attributes or field names of your Alfresco documents and comments to Amazon Q index field names.</td>
</tr>
<tr>
<td>• comment</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>aspectProperties</td>
<td>A list of specific 'Aspects' content that you want to index.</td>
</tr>
<tr>
<td>enableFineGrainedControl</td>
<td>true to crawl 'Aspects'.</td>
</tr>
<tr>
<td>isCrawlComment</td>
<td>true to index comments.</td>
</tr>
<tr>
<td>• inclusionFileNamePatterns</td>
<td>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.</td>
</tr>
</tbody>
</table>
## Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>exclusionFileNamePatterns</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>exclusionFileTypePatterns</code></td>
<td></td>
</tr>
<tr>
<td><code>exclusionFilePathPatterns</code></td>
<td></td>
</tr>
</tbody>
</table>

**type**

The type of data source. Specify ALFRESCO as your data source type.

**secretArn**

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:

```json
{
    "username": "username",
    "password": "password"
}
```

If using OAuth 2.0 authentication:

```json
{
    "clientId": "client ID",
    "clientSecret": "client secret",
    "tokenUrl": "token URL"
}
```
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index.</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>true to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to certain documents. See Identity crawler for more information.</td>
</tr>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>

### ACL crawling

When you connect an Alfresco data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Alfresco instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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're 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're mapped from the user emails as the IDs in Alfresco.
For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

### IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the `BatchPutDocument` and `BatchDeleteDocument` operations to ingest documents.
- Permission to access the [User Store](#) API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToGetSecret",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[{secret_id}]"
            ]
        },
        {
            "Sid": "AllowsAmazonQToDecryptSecret",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:{{region}}:{{account_id}}:cipher-text:
```
"arn:aws:kms:{region}:{account_id}:key/[[key_id]]",
"Condition": {
  "StringLike": {
    "kms:ViaService": [
      "secretsmanager.*.amazonaws.com"
    ]
  }
},
{
  "Sid": "AllowsAmazonQToIngestDocuments",
  "Effect": "Allow",
  "Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
  ],
  "Resource": "arn:aws:qbusiness:{region}:{source_account}:application/{application_id}/index/{index_id}"
},
{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroup"
  ],
  "Resource": [
    "arn:aws:qbusiness:{region}:{account_id}:application/{application_id}"
  ]
},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "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]]"
],

{
  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
  "Effect": "Allow",
  "Action": [  "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "AMAZON_Q"
      ]
    }
  }
},

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

{
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [  "ec2:CreateNetworkInterfacePermission"
  ]
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQServicePrincipal",
            "Effect": "Allow",
            "Principal": {
                "Service": "qbusiness.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "{{source_account}}"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
                }
            }
        }
    ]
}
```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting Alfresco (Server) to Amazon Q

Alfresco is a content management service (CMS) that helps customers store and manage their content. You can connect Alfresco (Server) instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

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 username 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.

• **Optional**: Generated OAuth 2.0 credentials in Alfresco. The credentials include client ID, client secret, and token URL. For more information about 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.

**In your AWS account, make sure you have:**

• Created an [IAM role](#) for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your Alfresco (Server) authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

**Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

**Using the console**

On the [Alfresco](#) page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Source** – Choose [Alfresco server](#).
   
   a. **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`.
   
   b. **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`. 
c. **SSL certificate location** – Enter the path to an SSL certificate file stored in an Amazon S3 bucket.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   ![Note](image)

   **Note**

   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

4. **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 username, and password.

   If you chose [OAuth 2.0 authentication](#), enter a name for the secret, client ID, client secret, and token URL.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).
8. In **Sync scope**, enter the following information:

a. **Content** – Choose whether to crawl content marked with 'Aspects' in Alfresco, content within a specific Alfresco site, or content across all your Alfresco sites.

b. **Additional configuration – optional** – 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.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   - **Full sync** – Sync all content regardless of the previous sync status.
   
   - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see Sync run schedule.

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See Tags for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the Data source details page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.
   
   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see Field mappings.

13. To finish connecting your data source to Amazon Q, select Add data source.

    You are taken to the Data source details, where you can view your data source configuration details.

14. In Data source details, choose Sync now to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.
Note

You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

Using the API

You use the CreateDataSource action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

Alfresco JSON schema

The following is the Alfresco JSON schema:

```json
{
   "$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"
},
"dateFormat": {
   "type": "string",
   "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
},
"required": [
   "indexFieldName",
   "indexFieldType",
   "dataSourceFieldName"
]
},
"comment": {
   "type": "object",
   "properties": {
      "fieldMappings": {
         "type": "array",
         "items": {
            "anyOf": [
               {
                  "type": "object",
                  "properties": {
                     "source": {
                        "type": "string"
                     },
                     "target": {
                        "type": "string"
                     },
                     "transform": {
                        "type": "string"  
                     }
               }
            ]  
         }
      }
   }
},
"required": [
   "fieldMappings"
]
}
"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": {
    "isCrawlAcl": {
      "type": "boolean"
    },
    "fieldForUserId": {
      "type": "string"
},
"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",
  "minLength": 20,
  "maxLength": 2048
}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>siteId</td>
<td>The identifier of the Alfresco site.</td>
</tr>
</tbody>
</table>
| 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
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>webAppUrl</td>
<td>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 <code>https://example.com</code>.</td>
</tr>
<tr>
<td>repositoryAdditionalProperties</td>
<td>Additional properties for content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>authType</td>
<td>The type of authentication that you use, whether OAuth2 or Basic.</td>
</tr>
<tr>
<td>type (deployment)</td>
<td>The type of Alfresco that you use, whether PAAS or ON-PREM.</td>
</tr>
<tr>
<td>crawlType</td>
<td>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).</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>• document</td>
<td>A list of objects that map the attributes or field names of your Alfresco documents and comments to Amazon Q index field names.</td>
</tr>
<tr>
<td>• comment</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>aspectProperties</td>
<td>A list of specific 'Aspects' content that you want to index.</td>
</tr>
<tr>
<td>enableFineGrainedControl</td>
<td>true to crawl 'Aspects'.</td>
</tr>
<tr>
<td>isCrawlComment</td>
<td>true to index comments.</td>
</tr>
<tr>
<td>• inclusionFileNamePatterns</td>
<td>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.</td>
</tr>
<tr>
<td>• inclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionFilePathPatterns</td>
<td></td>
</tr>
<tr>
<td>• exclusionFileNamePatterns</td>
<td>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.</td>
</tr>
<tr>
<td>• exclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• exclusionFilePathPatterns</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify ALFRESCO as your data source type.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</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:</td>
</tr>
<tr>
<td></td>
<td>If using basic authentication:</td>
</tr>
</tbody>
</table>
| | `{  
|   "username": "user name",  
|   "password": "password"  
| }  |
| | If using OAuth 2.0 authentication: |
| | `{  
|   "clientId": "client ID",  
|   "clientSecret": "client secret",  
|   "tokenUrl": "token URL"  
| }  |
| syncMode | Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options: |
| | • Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index. |
| | • Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index. |
**Configuration**

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enableIdentityCrawler</td>
<td>true to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to certain documents. See <a href="#">Identity crawler</a> for more information.</td>
</tr>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>

**ACL crawling**

When you connect an Alfresco data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Alfresco instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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're 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're mapped from the user emails as the IDs in Alfresco.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
• Permission to access the User Store API operations to ingest user and group access control information from documents.

• Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.

• Permission to access the SSL certificate stored in your Amazon S3 bucket.

• (Optional) If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowsAmazonQToGetS3Objects",
        "Action": [
            "s3:GetObject"
        ],
        "Resource": [
            "arn:aws:s3:::{{input_bucket_name}}/*"
        ],
        "Effect": "Allow",
        "Condition": {
            "StringEquals": {
                "aws:ResourceAccount": "{{account_id}}"
            }
        }
    },
    {
        "Sid": "AllowsAmazonQToGetSecret",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:GetSecretValue"
        ],
        "Resource": [
            "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
        ]
    },
    {
        "Sid": "AllowsAmazonQToDecryptSecret",
        "Effect": "Allow",
        "Action": [
            "kms:Decrypt"
        ],
        "Resource": [
```
"arn:aws:kms:{{region}}:{{account_id}}:key/[[key_id]]",

"Condition": {
    "StringLike": {
        "kms:ViaService": [
            "secretsmanager.*.amazonaws.com"
        ]
    }
}
},
{
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ],
    "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},
{
    "Sid": "AllowsAmazonQToIngestPrincipalMapping",
    "Effect": "Allow",
    "Action": [
        "qbusiness:PutGroup",
        "qbusiness:CreateUser",
        "qbusiness:DeleteGroup",
        "qbusiness:UpdateUser",
        "qbusiness:ListGroups"
    ],
    "Resource": [
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**
    ]
},
{
    "Sid": "AllowsAmazonQToCreateAndDeleteNI",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",

        "ec2:DeleteNetworkInterface",

        "ec2:DescribeNetworkInterfaces"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/{{network_interface_id}}"
}
"ec2:DeleteNetworkInterface",
"Resource": [
"arn:aws:ec2:{{region}}:{{account_id}}:subnet/[subnet_ids]",
"arn:aws:ec2:{{region}}:{{account_id}}:security-group/[[security_group]]"
],
{
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterface",
"ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringLike": {
"aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
},
"ForAllValues:StringEquals": {
"aws:TagKeys": [
"AMAZON_Q"
]
}
}
},
{
"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
"ec2:CreateTags"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringEquals": {
"ec2:CreateAction": "CreateNetworkInterface"
}
}
},
{
"Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringLike": {
"aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
},
"ForAllValues:StringEquals": {
"aws:TagKeys": [
"AMAZON_Q"
]
}
}
}
"Action": [
   "ec2:CreateNetworkInterfacePermission"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
   "StringLike": {
      "aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_{{*}}"
   }
},
"Sid": "AllowsAmazonQToDescribeResourcesForVPC",
"Effect": "Allow",
"Action": [
   "ec2:DescribeNetworkInterfaces",
   "ec2:DescribeAvailabilityZones",
   "ec2:DescribeNetworkInterfaceAttribute",
   "ec2:DescribeVpcs",
   "ec2:DescribeRegions",
   "ec2:DescribeNetworkInterfacePermissions",
   "ec2:DescribeSubnets"
],
"Resource": "*
"
"

To allow Amazon Q to assume a role, you must also use the following trust policy:

{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
      "Effect": "Allow",
      "Principal": {
         "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
         "StringEquals": {
            "aws:SourceAccount": "{{source_account}}"
         }
      }
   }
]
For more information on Amazon Q data source connector IAM roles, see [IAM roles for Amazon Q data source connectors](#).

## Connecting Aurora (MySQL) to Amazon Q

Aurora (MySQL) is a relational database management system (RDBMS) built for the cloud. You can connect your Aurora (MySQL) instance to Amazon Q—using either the AWS Management Console, CLI, or the [CreateDataSource](#) API—and create an Amazon Q web experience.

The Amazon Q Aurora (MySQL) data source connector supports Aurora MySQL 3 and Aurora Serverless MySQL 8.0.

### Learn more

- For an overview of the Amazon Q web experience creation process, see [Configuring an application](#).
- For an overview of connector features, see [Data source connector concepts](#).
- For information about connector configuration best practices, see [Connector configuration best practices](#).

### Topics

- [Prerequisites](#)
- Using the console
- Using the API
- ACL crawling
- [IAM roles](#)
Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Aurora (MySQL), make sure you have:

• Noted your database user name and password.

⚠️ Important
As a best practice, provide Amazon Q with read-only database credentials.

• Copied your database host URL, port, and instance. You can find this information on the Amazon RDS console.

In your AWS account, make sure you have:

• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your Aurora (MySQL) authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

⚠️ Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Aurora (MySQL) page, enter the following information:

1. Name – Name your data source for easy tracking.
   
   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In Source, enter the following information:
Amazon Q

a. **Host** – Enter the database host URL, for example: 
   \[http://instance URL .region .rds.amazonaws.com\].

b. **Port** – Enter the database port, for example, 5432.

c. **Instance** – Enter the database instance, for example postgres.

d. **SSL certificate location** – Choose to enter the Amazon S3 path to your SSL certificate file.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   ![Note]
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

4. In **Authentication**, enter the following information for your **AWS Secrets Manager secret**.

   a. **Secret name** – A name for your secret.

   b. For **Database user name**, and **Password** – Enter the authentication credential values you copied from your database.

   c. Choose **Save**.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).
7. In **Sync scope**, enter the following information:

- **SQL query** – Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Q 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 names of the document body column within your database table.

8. In **Additional configuration – optional** – Configure the following settings:

- **Change-detecting columns** – Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q 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 Q 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.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

- **Full sync** – Sync all content regardless of the previous sync status.

- **New or modified content sync** – Sync only new and modified documents.

- **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.
For more details, see Sync mode.

10. **In Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see Sync run schedule.

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See Tags for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the Data source details page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see Field mappings.

13. To finish connecting your data source to Amazon Q, select **Add data source**.

You are taken to the Data source details, where you can view your data source configuration details.

14. In Data source details, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

> **Note**
>
> You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

### Using the API

You use the CreateDataSource action to connect a data source to your Amazon Q application.
Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

JSON schema

The following is the JSON schema:

```json
{
    "$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": [  
  "fieldMappings"
  
},
"required": [  
  ]
},
"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"
    },
    "serverlessAurora": {
      "type": "string",
      "Aurora (MySQL)
    },
    "serverlessAurora": {
      "type": "string",
      "Aurora (MySQL)
    }
  }
}
}
The following table provides information about important JSON keys to configure.
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>Required configuration information for connecting your data source.</td>
</tr>
<tr>
<td></td>
<td>• dbType—The type of Java database you are using, whether mysql, db2, postgresql, oracle, or sqlserver.</td>
</tr>
<tr>
<td></td>
<td>• dbHost—The database host name.</td>
</tr>
<tr>
<td></td>
<td>• dbPort—The database port.</td>
</tr>
<tr>
<td></td>
<td>• dbInstance—The database instance.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>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.</td>
</tr>
<tr>
<td>document</td>
<td>A list of objects that map the attributes or field names of your database content to Amazon Q index field names. For more information, see Mapping data source fields.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.</td>
</tr>
<tr>
<td>primaryKey</td>
<td>Provide the primary key for the database table. This identifies a table within your database.</td>
</tr>
<tr>
<td>titleColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>bodyColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>sqlQuery</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>timestampColumn</td>
<td>Enter the name of the column which contains time stamps. Amazon Q uses time stamp information to detect changes in your content and sync only changed content.</td>
</tr>
<tr>
<td>timestampFormat</td>
<td>Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.</td>
</tr>
<tr>
<td>timezone</td>
<td>Enter the name of the column which contains time zones for the content to be crawled.</td>
</tr>
<tr>
<td>changeDetectingColumns</td>
<td>Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q will re-index content when there is a change in any of these columns</td>
</tr>
<tr>
<td>allowedUsersColumns</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>allowedGroupsColumn</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>sourceURIColumn</td>
<td>Enter the name of the column which contains Source URLs to be indexed.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>isSslEnabled</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify JDBC as your data source type.</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose</td>
</tr>
<tr>
<td></td>
<td>• FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index</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:</td>
</tr>
<tr>
<td></td>
<td>```</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;user name&quot;: &quot;database user name&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;password&quot;: &quot; password&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>
|                 | ```
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>The version of the template that is currently supported.</td>
</tr>
</tbody>
</table>

### ACL crawling

When you connect a database data source to Amazon Q, Amazon Q crawls user and group information from a column in the source table. You specify this column in the console or using the configuration parameter as part of the `CreateDataSource` operation.

If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

### IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the `BatchPutDocument` and `BatchDeleteDocument` operations to ingest documents.
- Permission to access the [User Store](#) API operations to ingest user and group access control information from documents.
• Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.

• Permission to access the SSL certificate stored in your Amazon S3 bucket.

• *(Optional)* If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
   "Version": "2012-10-17",
   "Statement": [{
       "Sid": "AllowsAmazonQToGetS3Objects",
       "Action": [
           "s3:GetObject"
       ],
       "Resource": [
           "arn:aws:s3:::{{input_bucket_name}}/*"
       ],
       "Effect": "Allow",
       "Condition": {
           "StringEquals": {
               "aws:ResourceAccount": "{{account_id}}"
           }
       }
   },
   {"Sid": "AllowsAmazonQToGetSecret",
   "Effect": "Allow",
   "Action": [
       "secretsmanager:GetSecretValue"
   ],
   "Resource": [
       "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
   ]},
   { "Sid": "AllowsAmazonQToDecryptSecret",
   "Effect": "Allow",
   "Action": [
       "kms:Decrypt"
   ],
   "Resource": [
       "arn:aws:kms:{{region}}:{{account_id}}:key:[[key_id]]"
   ]},
   {"Sid": "AllowsAmazonQToDecryptSecret",
   "Effect": "Allow",
   "Action": [
       "kms:Decrypt"
   ],
   "Resource": [
       "arn:aws:kms:{{region}}:{{account_id}}:key:[[key_id]]"
   ]}
}
"Condition": {
    "StringLike": {
        "kms:ViaService": [
            "secretsmanager.*.amazonaws.com"
        ]
    }
},
{
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ],
    "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},
{
    "Sid": "AllowsAmazonQToIngestPrincipalMapping",
    "Effect": "Allow",
    "Action": [
        "qbusiness:PutGroup",
        "qbusiness:CreateUser",
        "qbusiness:DeleteGroup",
        "qbusiness:UpdateUser",
        "qbusiness:ListGroups"
    ],
    "Resource": [
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**"
    ]
},
{
    "Sid": "AllowsAmazonQToCreateAndDeleteNI",
    "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]]"
],
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/********",
  "Condition": {
    "StringLike": {
      "aws:RequestTag/AMAZON_Q":
        "qbusiness_{{account_id}}_{{application_id}}_**"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "AMAZON_Q"
      ]
    }
  }
},
{
  "Sid": "AllowsAmazonQToCreateTags",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/********",
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
    }
  }
},
{
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ]
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
            "Effect": "Allow",
            "Principal": {
                "Service": "qbusiness.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "{{source_account}}"
                },
                "ArnLike": {

```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

**Connecting Aurora (PostgreSQL) to Amazon Q**

Aurora (PostgreSQL) is a relational database management system (RDBMS) built for the cloud. You can connect your Aurora (PostgreSQL) instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

The Amazon Q Aurora (PostgreSQL) data source connector supports Aurora PostgreSQL 1.

**Learn more**

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

**Topics**

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

**Prerequisites**

Before you begin, make sure that you have completed the following prerequisites.
In Aurora (PostgreSQL), make sure you have:

- Noted your database user name and password.

⚠️ **Important**

As a best practice, provide Amazon Q with read-only database credentials.

- Copied your database host URL, port, and instance. You can find this information on the Amazon RDS console.

In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your Aurora (PostgreSQL) authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

✅ **Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

**Using the console**

On the Aurora (PostgreSQL) page, enter the following information:

1. **Name** – Name your data source for easy tracking.

   ✅ **Note**: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In **Source**, enter the following information:

   a. **Host** – Enter the database host URL, for example: `http://instance URL.region.rds.amazonaws.com`.

   b. **Port** – Enter the database port, for example, 5432.
c. **Instance** – Enter the database instance, for example `postgres`.

d. **Enable SSL certificate location** – Choose to enter the Amazon S3 path to your SSL certificate file.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   ![](image)

   **Note**

   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

4. In **Authentication** – Enter the following information for your AWS Secrets Manager secret.

   a. **Secret name** – A name for your secret.

   b. For **Database user name**, and **Password** – Enter the authentication credential values you copied from your database.

   c. Choose **Save**.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

7. In **Sync scope**, enter the following information:
• **SQL query** – Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Q 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 names of the document body column within your database table.

8. In **Additional configuration – optional** – Configure the following settings:

• **Change-detecting columns** – Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q 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 Q 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.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

• **Full sync** – Sync all content regardless of the previous sync status.

• **New or modified content sync** – Sync only new and modified documents.

• **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.
10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see **Sync run schedule**.

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See **Tags** for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see **Field mappings**.

13. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

   **Note**
   
   You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the **CreateDataSource** action to connect a data source to your Amazon Q application.
Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**JSON schema**

The following is the JSON schema:

```json
{
  "$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"
]
```

Aurora (PostgreSQL) 218
"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"
    },
    "serverlessAurora": {
      "type": "string"
    }
  }
},
"Aurora (PostgreSQL)"
The following table provides information about important JSON keys to configure.
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>Required configuration information for connecting your data source.</td>
</tr>
<tr>
<td></td>
<td>• <code>dbType</code>—The type of Java database you are using, whether <code>mysql</code>, <code>db2</code>, <code>postgresql</code>, <code>oracle</code>, or <code>sqlserver</code>.</td>
</tr>
<tr>
<td></td>
<td>• <code>dbHost</code>—The database host name.</td>
</tr>
<tr>
<td></td>
<td>• <code>dbPort</code>—The database port.</td>
</tr>
<tr>
<td></td>
<td>• <code>dbInstance</code>—The database instance.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>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.</td>
</tr>
<tr>
<td>document</td>
<td>A list of objects that map the attributes or field names of your database content to Amazon Q index field names. For more information, see <a href="#">Mapping data source fields</a>.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.</td>
</tr>
<tr>
<td>primaryKey</td>
<td>Provide the primary key for the database table. This identifies a table within your database.</td>
</tr>
<tr>
<td>titleColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>bodyColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>sqlQuery</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>timestampColumn</td>
<td>Enter the name of the column which contains time stamps. Amazon Q uses time stamp information to detect changes in your content and sync only changed content.</td>
</tr>
<tr>
<td>timestampFormat</td>
<td>Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.</td>
</tr>
<tr>
<td>timezone</td>
<td>Enter the name of the column which contains time zones for the content to be crawled.</td>
</tr>
<tr>
<td>changeDetectingColumns</td>
<td>Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q will re-index content when there is a change in any of these columns.</td>
</tr>
<tr>
<td>allowedUsersColumns</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>allowedGroupsColumn</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>sourceURIColumn</td>
<td>Enter the name of the column which contains Source URLs to be indexed.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>isSslEnabled</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify JDBC as your data source type.</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose</td>
</tr>
<tr>
<td></td>
<td>• FORCED_FULL_CRAWL  to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• FULL_CRAWL         to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• CHANGE_LOG         to incrementally crawl only new and modified content each time your data source syncs with your index.</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:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;user name&quot;: &quot;database user name&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;password&quot;: &quot;password&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>The version of the template that is currently supported.</td>
</tr>
</tbody>
</table>

### ACL crawling

When you connect a database data source to Amazon Q, Amazon Q crawls user and group information from a column in the source table. You specify this column in the console or using the configuration parameter as part of the `CreateDataSource` operation.

If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

### IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the [User Store](#) API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.

- Permission to access the SSL certificate stored in your Amazon S3 bucket.

- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToGetS3Objects",
            "Action": ["s3:GetObject"],
            "Resource": ["arn:aws:s3:::{{input_bucket_name}}/*"],
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "{{account_id}}"
                }
            }
        },
        {
            "Sid": "AllowsAmazonQToGetSecret",
            "Effect": "Allow",
            "Action": ["secretsmanager:GetSecretValue"],
            "Resource": ["arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"]
        },
        {
            "Sid": "AllowsAmazonQToDecryptSecret",
            "Effect": "Allow",
            "Action": ["kms:Decrypt"],
            "Resource": ["arn:aws:kms:{{region}}:{{account_id}}:key:[[key_id]]"]
        }
    ]
}
```
"Condition": {
  "StringLike": {
    "kms:ViaService": [
      "secretsmanager.*.amazonaws.com"
    ]
  }
},
{
  "Sid": "AllowsAmazonQToIngestDocuments",
  "Effect": "Allow",
  "Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
  ],
  "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},
{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroups"
  ],
  "Resource": [
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**
  ]
},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "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]]"
],  
{  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",  "Effect": "Allow",  "Action": [  "ec2:CreateNetworkInterface",  "ec2:DeleteNetworkInterface"
],  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",  "Condition": {  "StringLike": {  "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"  },  "ForAllValues:StringEquals": {  "aws:TagKeys": [  "AMAZON_Q"
  ]
  }  }
},  
{  "Sid": "AllowsAmazonQToCreateTags",  "Effect": "Allow",  "Action": [  "ec2:CreateTags"
],  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",  "Condition": {  "StringEquals": {  "ec2:CreateAction": "CreateNetworkInterface"
  }
  }
},  
{  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",  "Effect": "Allow",  "Action": [  "ec2:CreateNetworkInterfacePermission"
  ]
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnLike": {
        }
      }
    }
  ]
}
```
"aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
]
]
]
]

For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting Amazon FSx (Windows) to Amazon Q

Amazon FSx (Windows) is a fully managed, cloud based file server system that offers shared storage capabilities. You can connect your Amazon FSx (Windows) instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

The Amazon Q Amazon FSx (Windows) data source connector supports only Amazon FSx for Windows.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles
Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Amazon FSx (Windows), make sure you have:

- An Amazon FSx (Windows) account with read and mounting permissions.
- Noted your Amazon FSx authentication credentials for an Active Directory user account. This includes your Active Directory user name and your Domain Name System (DNS) domain name. For example, user@corp.example.com.
- Copied your Amazon FSx file system ID.
- Used an Amazon VPC (AWS VPC) where your Amazon FSx resides.

In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your Amazon FSx (Windows) authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Amazon FSx (Windows) page, enter the following information:

1. Name – Name your data source for easy tracking.
   
   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In Source, for Amazon FSx file system ID—Select your file system ID or create a new directory.
   
   Only already created file system IDs are displayed and available to connect.
3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   ![Note]

   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

4. **Authentication** – Enter the following information for your [AWS Secrets Manager secret](#).

   a. **Secret name**—A name for your secret.
   b. **For User name**—Enter the user name for Amazon FSx Active Directory account.
   c. **For Password**—Enter the password for the Amazon FSx Active Directory account.
   d. Choose Save.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

8. In **Sync scope**, enter the following information:

   - **Regex patterns**—Add regular expression patterns to include or exclude certain content. You can add up to 100 patterns.
9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

   For more details, see [Sync mode](#).

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.

    You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

    **Note**

    You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to
view your CloudWatch logs for a data source sync job in progress, it can be because the
CloudWatch logs are not available yet. Wait for some time and check again.

Using the API

You use the `CreateDataSource` action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other
configuration information specific to your data source connector.

Amazon FSx JSON schema

The following is the Amazon FSx JSON schema:

```json
{
  "$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"
  ]}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>fileSystemId</td>
<td>The identifier of the Amazon FSx file system. You can find your file system ID on the FileSystems dashboard in the Amazon FSx console.</td>
</tr>
<tr>
<td>fileSystemType</td>
<td>The type of Amazon FSx you use: ONTAP.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• All</td>
<td>A list of objects that map the attributes or field names of your Amazon FSx pages and assets to Amazon Q index field names.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>• inclusionPatterns</td>
<td>A list of regular expression patterns to include specific content from your Amazon FSx data source. Content that match 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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>exclusionPatterns</td>
<td>A list of regular expression patterns to exclude specific content from your Amazon FSx data source. Content that match 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.</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td><code>true</code> to activate identity crawler. Identity crawler is activated by default. Crawling identity information on users and groups with access to specific documents is useful for user context filtering. Search results are filtered based on the user or their group access to documents. See <a href="Identitycrawler">Identity crawler</a> for more information.</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify FSX as your data source type.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>

### ACL crawling

When you connect an Amazon FSx data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from the directory service of the Amazon FSx instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

The 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.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

### IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the `BatchPutDocument` and `BatchDeleteDocument` operations to ingest documents.
- Permission to access the [User Store](#) API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
• Permission to access the SSL certificate stored in your Amazon S3 bucket.

• *(Optional)* If you’re using Amazon VPC, permission to access your Amazon VPC.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToGetS3Objects",
            "Action": [ "s3:GetObject" ],
            "Resource": [ "arn:aws:s3:::{input_bucket_name}/*" ],
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "{{account_id}}"
                }
            }
        },
        {
            "Sid": "AllowsAmazonQToGetSecret",
            "Effect": "Allow",
            "Action": [ "secretsmanager:GetSecretValue" ],
            "Resource": [ "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]" ]
        },
        {
            "Sid": "AllowsAmazonQToDecryptSecret",
            "Effect": "Allow",
            "Action": [ "kms:Decrypt" ],
            "Resource": [ "arn:aws:kms:{{region}}:{{account_id}}:key/[{{key_id}}]" ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [ "AmazonFSx" ]
                }
            }
        }
    ]
}
```
"secretsmanager.*.amazonaws.com"

],

}

},

{
  "Sid": "AllowsAmazonQToIngestDocuments",
  "Effect": "Allow",
  "Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
  ],
  "Resource": "arn:aws:qbusiness:{region}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},

{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroups"
  ],
  "Resource": [
    "arn:aws:qbusiness:{region}:{{account_id}}:application/{{application_id}}",
    "arn:aws:qbusiness:{region}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
    "arn:aws:qbusiness:{region}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/*"
  ]
},

{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
  ],
  "Resource": [
    "arn:aws:ec2:{region}:{{account_id}}:subnet/[[subnet_ids]]",
    "arn:aws:ec2:{region}:{{account_id}}:subnet/[[subnet_ids]]/associate-ipv6-address"
  ]
}
"arn:aws:ec2:{{region}}:{{account_id}}:security-group/
[[security_group]]"
],
{
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterface",
"ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*",
"Condition": {
"StringLike": {
"aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
},
"ForAllValues:StringEquals": {
"aws:TagKeys": [
"AMAZON_Q"
]
}
}
},
{
"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
"ec2:CreateTags"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*",
"Condition": {
"StringEquals": {
"ec2:CreateAction": "CreateNetworkInterface"
}
}
},
{
"Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterfacePermission"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*"
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
        }
      }
    }
  ]
}
```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q
data source connectors.

Connecting Amazon RDS (Microsoft SQL Server) to Amazon Q

Amazon RDS (Microsoft SQL Server) is a relational database management system (RDBMS) built for
the cloud. You can connect your Amazon RDS (Microsoft SQL Server) instance to Amazon Q – using
either the AWS Management Console, CLI, or the CreateDataSource API – and create an Amazon Q
web experience.

The Amazon Q Aurora (MySQL) data source connector supports Microsoft SQL Server 2019.

Learn more

• For an overview of the Amazon Q web experience creation process, see Configuring an
  application.
• For an overview of connector features, see Data source connector concepts.
• For information about connector configuration best practices, see Connector configuration best
  practices.

Topics

• Prerequisites
• Using the console
• Using the API
• ACL crawling
• IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Amazon RDS (Microsoft SQL Server), make sure you have:
• Noted your database user name and password.

⚠️ Important
As a best practice, provide Amazon Q with read-only database credentials.

• Copied your database host URL, port, and instance.

In your AWS account, make sure you have:

• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
• Stored your Amazon RDS (Microsoft SQL Server) authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

🔍 Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Amazon RDS (Microsoft SQL Server) page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In **Source**, enter the following information:
   a. **Host** – Enter the database host name.
   b. **Port** – Enter the database port.
   c. **Instance** – Enter the database instance.
   d. **SSL certificate location** – Choose to enter the Amazon S3 path to your SSL certificate file.
3. In **Authentication** – Enter the following information for your AWS Secrets Manager secret.
a. **Secret name** – A name for your secret.

b. For **Database user name**, and **Password** – Enter the authentication credential values you copied from your database.

c. Choose **Save**.

4. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

5. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

6. In **Sync scope**, enter the following information:

   - **SQL query** – Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Q 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 names of the document body column within your database table.

7. In **Additional configuration – optional** – Configure the following settings:

   - **Change-detecting columns** – Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q 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 Q 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.

8. **In Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   • **Full sync** – Sync all content regardless of the previous sync status.
   
   • **New or modified content sync** – Sync only new and modified documents.
   
   • **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

   For more details, see [Sync mode](#).

9. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

10. **In Sync run schedule, for Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

11. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the [Data source details](#) page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.
   
   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).
12. To finish connecting your data source to Amazon Q, select **Add data source**.

You are taken to the **Data source details**, where you can view your data source configuration details.

13. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

**Note**

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the **CreateDataSource** action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**JSON schema**

The following is the JSON schema:

```json
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "properties": {
      "connectionConfiguration": {
         "type": "object",
         "properties": {
            "repositoryEndpointMetadata": {
               "type": "object",
               "properties": {
                  "dbType": {
                     "type": "string",
                     "enum": ["mysql", "Amazon RDS (Microsoft SQL Server)"
                  ]
               }
            }
         }
      }
   }
}
```
"db2",
"postgresql",
"oracle",
"sqlserver"
],
"dbHost": {
  "type": "string"
},
"dbPort": {
  "type": "string"
},
"dbInstance": {
  "type": "string"
},
"required": [
  "dbType",
  "dbHost",
  "dbPort",
  "dbInstance"
],
"repositoryEndpointMetadata": {
  "type": "string"
},
"repositoryConfigurations": {
  "type": "object",
  "properties": {
    "document": {
      "type": "object",
      "properties": {
        "fieldMappings": {
          "type": "array",
          "items": [
            {
              "type": "object",
              "properties": {
                "indexFieldName": {
                  "type": "string"
                },
                "indexFieldType": {
                  "type": "string"
                }
              }
            }
          ]
        }
      }
    }
  }
}
"type": "string"
},
"dataSourceFieldName": {
  "type": "string"
}
],
"required": [
  "indexFieldName",
  "indexFieldType",
  "dataSourceFieldName"
]
}
]
}
"required": [ ]
}
],
"additionalProperties": {
  "type": "object",
  "properties": {
    "primaryKey": {
      "type": "string"
    },
    "titleColumn": {
      "type": "string"
    },
    "bodyColumn": {
      "type": "string"
    },
    "sqlQuery": {
      "type": "string",
      "not": {
        "pattern": ";+"
      }
    },
    "timestampColumn": {
      "type": "string"
    }
  }
},
"Amazon RDS (Microsoft SQL Server)"
249
"timestampFormat": {
    "type": "string"
},
"timezone": {
    "type": "string"
},
"changeDetectingColumns": {
    "type": "array",
    "items": {
        "type": "string"
    }
},
"allowedUsersColumn": {
    "type": "string"
},
"allowedGroupsColumn": {
    "type": "string"
},
"sourceURIColumn": {
    "type": "string"
},
"serverlessAurora": {
    "type": "string",
    "enum": ["true", "false"]
},
"required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
},
"type": {
    "type": "string",
    "pattern": "JDBC"
},
"syncMode": {
    "type": "string",
    "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL"
    ]
},
"secretArn": {
    "type": "string"
}
},
"version": {
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>Required configuration information for connecting your data source.</td>
</tr>
<tr>
<td></td>
<td>• dbType—The type of Java database you are using, whether mysql, db2, postgresql, oracle, or sqlserver.</td>
</tr>
<tr>
<td></td>
<td>• dbHost—The database host name.</td>
</tr>
<tr>
<td></td>
<td>• dbPort—The database port.</td>
</tr>
<tr>
<td></td>
<td>• dbInstance—The database instance.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>document</td>
<td>A list of objects that map the attributes or field names of your database content to Amazon Q index field names. For more information, see <a href="#">Mapping data source fields</a>.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.</td>
</tr>
<tr>
<td>primaryKey</td>
<td>Provide the primary key for the database table. This identifies a table within your database.</td>
</tr>
<tr>
<td>titleColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>bodyColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>sqlQuery</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>timestampColumn</td>
<td>Enter the name of the column which contains time stamps. Amazon Q uses time stamp information to detect changes in your content and sync only changed content.</td>
</tr>
<tr>
<td>timestampFormat</td>
<td>Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.</td>
</tr>
<tr>
<td>timezone</td>
<td>Enter the name of the column which contains time zones for the content to be crawled.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>changeDetectingColumns</td>
<td>Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q will re-index content when there is a change in any of these columns.</td>
</tr>
<tr>
<td>allowedUsersColumns</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>allowedGroupsColumn</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>sourceURIColumn</td>
<td>Enter the name of the column which contains Source URLs to be indexed.</td>
</tr>
<tr>
<td>isSslEnabled</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify JDBC as your data source type.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose</td>
</tr>
<tr>
<td></td>
<td>• FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index.</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:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;user name&quot;: &quot;database user name&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;password&quot;: &quot;password&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>version</td>
<td>The version of the template that is currently supported.</td>
</tr>
</tbody>
</table>

**ACL crawling**

When you connect a database data source to Amazon Q, Amazon Q crawls user and group information from a column in the source table. You specify this column in the console or using the configuration parameter as part of the CreateDataSource operation.
If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- Authorization
- Identity crawler
- Understanding User Store

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- Permission to access the SSL certificate stored in your Amazon S3 bucket.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
   "Version": "2012-10-17",
   "Statement": [{
      "Sid": "AllowsAmazonQToGetS3Objects",
      "Action": ["s3:GetObject"
   ],
```

Amazon RDS (Microsoft SQL Server)
"Resource": [  "arn:aws:s3:::{{input_bucket_name}}/*"
],
"Effect": "Allow",
"Condition": {  "StringEquals": {  "aws:ResourceAccount": "{{account_id}}"
  }
}
},
{
"Sid": "AllowsAmazonQToGetSecret",
"Effect": "Allow",
"Action": [  "secretsmanager:GetSecretValue"
],
"Resource": [  "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:{{secret_id}}"
]
},
{
"Sid": "AllowsAmazonQToDecryptSecret",
"Effect": "Allow",
"Action": [  "kms:Decrypt"
],
"Resource": [  "arn:aws:kms:{{region}}:{{account_id}}:key:{{key_id}}"
],
"Condition": {  "StringLike": {  "kms:ViaService": [  "secretsmanager.*.amazonaws.com"
 ]
  }
}
},
{
"Sid": "AllowsAmazonQToIngestDocuments",
"Effect": "Allow",
"Action": [  "qbusiness:BatchPutDocument",
  "qbusiness:BatchDeleteDocument"  
]  }
"Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}",

{
    "Sid": "AllowsAmazonQToIngestPrincipalMapping",
    "Effect": "Allow",
    "Action": [
        "qbusiness:PutGroup",
        "qbusiness:CreateUser",
        "qbusiness:DeleteGroup",
        "qbusiness:UpdateUser",
        "qbusiness:ListGroups"
    ],
    "Resource": [
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/*"
    ]
},
{
    "Sid": "AllowsAmazonQToCreateAndDeleteNI",
    "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]]"
    ]
},
{
    "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
    "Condition": {

"StringLike": {
  "aws:RequestTag/AMAZON_Q": "qbusiness_{account_id}_{application_id}#*"
  
  "ForAllValues:_EQUALS": {
  "aws:TagKeys": [ "AMAZON_Q"
  ],
  } }
},

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

{ "Sid": "AllowsAmazonQToDescribeResourcesForVPC", "Effect": "Allow", "Action": [ "ec2:DescribeNetworkInterfaces",
  "ec2:DescribeAvailabilityZones",
  "ec2:DescribeNetworkInterfaceTags" ] }
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
        }
      }
    }
  ]
}
```

For more information on Amazon Q data source connector IAM roles, see [IAM roles for Amazon Q data source connectors](#).

**Connecting Amazon RDS (MySQL) to Amazon Q**

Amazon RDS (MySQL) (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. You can connect your Amazon RDS (MySQL) instances to Amazon Q. To do so, you need to:

1. Create a IAM role for your Amazon Q data source connector.
2. Add permissions to the role.
3. Configure the trust policy to allow Amazon Q to assume the role.
4. Associate the role with your Amazon Q data source connector.

For more information, see the Amazon RDS (MySQL) documentation.
RDS (MySQL) instance to Amazon Q – using either the AWS Management Console, CLI, or the CreateDataSource API – and create an Amazon Q web experience.

The Amazon Q Aurora (MySQL) data source connector supports Amazon RDS MySQL 5.6, 5.7, and 8.0.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Amazon RDS (MySQL), make sure you have:

- Noted your database user name and password.

⚠️ Important
As a best practice, provide Amazon Q with read-only database credentials.

- Copied your database host URL, port, and instance. You can find this information on the Amazon RDS console.

In your AWS account, make sure you have:
• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your Amazon RDS (MySQL) authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Amazon RDS (MySQL) page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Source**, enter the following information:
   
   a. **Host** – Enter the database host URL, for example: http://instance
      
      \_URL\_.region.rds.amazonaws.com.
   
   b. **Port** – Enter the database port, for example, 5432.
   
   c. **Instance** – Enter the database instance, for example postgres.
   
   d. **SSL certificate location** – Choose to enter the Amazon S3 path to your SSL certificate file.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.

Note
Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).
4. In Authentication – Enter the following information for your AWS Secrets Manager secret.
   a. **Secret name** – A name for your secret.
   b. For **Database user name**, and **Password** – Enter the authentication credential values you copied from your database.
   c. Choose **Save**.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:
   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

7. In **Sync scope**, enter the following information:
   - **SQL query** – Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Q 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 names of the document body column within your database table.

8. In **Additional configuration – optional** – Configure the following settings:
   - **Change-detecting columns** – Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q 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 Q 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.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   • **Full sync** – Sync all content regardless of the previous sync status.
   
   • **New or modified content sync** – Sync only new and modified documents.
   
   • **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

   For more details, see **Sync mode**.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see **Sync run schedule**.

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See **Tags** for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

    a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.
    
    b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.
For more information, see Field mappings.

13. To finish connecting your data source to Amazon Q, select Add data source.

You are taken to the Data source details, where you can view your data source configuration details.

14. In Data source details, choose Sync now to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

Note
You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

Using the API

You use the CreateDataSource action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

JSON schema

The following is the JSON schema:

```json
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "dbType": {
              "type": "object",
            }
          }
        }
      }
    }
  }
}
```
"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"
                }
              }
            }
          ]
        }
      }
    }
  }
}
}
Amazon Q

Developer Guide

"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": ";+"
}
},

Amazon RDS (MySQL)

266


"timestampColumn": {
  "type": "string"
},
"timestampFormat": {
  "type": "string"
},
"timezone": {
  "type": "string"
},
"changeDetectingColumns": {
  "type": "array",
  "items": {
    "type": "string"
  }
},
"allowedUsersColumn": {
  "type": "string"
},
"allowedGroupsColumn": {
  "type": "string"
},
"sourceURIColumn": {
  "type": "string"
},
"serverlessAurora": {
  "type": "string",
  "enum": ["true", "false"]
},
"required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
},
"type": {
  "type": "string",
  "pattern": "JDBC"
},
"syncMode": {
  "type": "string",
  "enum": [
    "FORCED_FULL_CRAWL",
    "FULL_CRAWL"
  ]
},
"secretArn": {
  "type": "string"
}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>Required configuration information for connecting your data source.</td>
</tr>
<tr>
<td></td>
<td>• dbType—The type of Java database you are using, whether mysql, db2,</td>
</tr>
<tr>
<td></td>
<td>postgresql, oracle, or sqlserver.</td>
</tr>
<tr>
<td></td>
<td>• dbHost—The database host name.</td>
</tr>
<tr>
<td></td>
<td>• dbPort—The database port.</td>
</tr>
<tr>
<td></td>
<td>• dbInstanceId—The database instance.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example,</td>
</tr>
<tr>
<td></td>
<td>configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>document</td>
<td>A list of objects that map the attributes or field names of your database content to Amazon Q index field names. For more information, see <a href="#">Mapping data source fields</a>.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.</td>
</tr>
<tr>
<td>primaryKey</td>
<td>Provide the primary key for the database table. This identifies a table within your database.</td>
</tr>
<tr>
<td>titleColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>bodyColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>sqlQuery</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>timestampColumn</td>
<td>Enter the name of the column which contains time stamps. Amazon Q uses time stamp information to detect changes in your content and sync only changed content.</td>
</tr>
<tr>
<td>timestampFormat</td>
<td>Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>timezone</td>
<td>Enter the name of the column which contains time zones for the content to be crawled.</td>
</tr>
<tr>
<td>changeDetectingColumns</td>
<td>Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q will re-index content when there is a change in any of these columns.</td>
</tr>
<tr>
<td>allowedUsersColumns</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>allowedGroupsColumn</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>sourceURIColumn</td>
<td>Enter the name of the column which contains Source URLs to be indexed.</td>
</tr>
<tr>
<td>isSslEnabled</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify JDBC as your data source type.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| syncMode     | Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose  
- FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index  
- FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index  
- CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index. |
| 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. |

**ACL crawling**

When you connect a database data source to Amazon Q, Amazon Q crawls user and group information from a column in the source table. You specify this column in the console or using the configuration parameter as part of the CreateDataSource operation.
If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- Authorization
- Identity crawler
- Understanding User Store

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- Permission to access the SSL certificate stored in your Amazon S3 bucket.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowsAmazonQToGetS3Objects",
    "Action": ["s3:GetObject"
  ]
}
```
"Resource": [
    "arn:aws:s3:::{{input_bucket_name}}/*"
],
"Effect": "Allow",
"Condition": {
    "StringEquals": {
        "aws:ResourceAccount": "{{account_id}}"
    }
}
},
{
    "Sid": "AllowsAmazonQToGetSecret",
    "Effect": "Allow",
    "Action": [
        "secretsmanager:GetSecretValue"
    ],
    "Resource": [
        "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:{{secret_id}}"
    ]
},
{
    "Sid": "AllowsAmazonQToDecryptSecret",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:{{region}}:{{account_id}}:key:{{key_id}}"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
                "secretsmanager.*.amazonaws.com"
            ]
        }
    }
}
},
{
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ]
}
"Resource": "arn:aws:qbusiness:{region}::{source_account}::application/{application_id}/index/{index_id}"
],
{
"Sid": "AllowsAmazonQToIngestPrincipalMapping",
"Effect": "Allow",
"Action": [
  "qbusiness:PutGroup",
  "qbusiness:CreateUser",
  "qbusiness:DeleteGroup",
  "qbusiness:UpdateUser",
  "qbusiness:ListGroups"
],
"Resource": [
  "arn:aws:qbusiness:{region}::{account_id}::application/{application_id}"
]
},
{
"Sid": "AllowsAmazonQToCreateAndDeleteNI",
"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]"
]
},
{
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
  "ec2:CreateNetworkInterface",
  "ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{region}::{account_id}::network-interface/*",
"Condition": {
"StringLike": {
    "aws:RequestTag/AMAZON_Q":
    "qbusiness_{account_id}_{application_id}*"
},
"ForAllValues:StringEquals": {
    "aws:TagKeys": [
    "AMAZON_Q"
    ]
}
},
{
    "Sid": "AllowsAmazonQToCreateTags",
    "Effect": "Allow",
    "Action": [
    "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:{region}:{account_id}:network-interface/**",
    "Condition": {
    "StringEquals": {
    "ec2:CreateAction": "CreateNetworkInterface"
    }
    }
},
{
    "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
    "Effect": "Allow",
    "Action": [
    "ec2:CreateNetworkInterfacePermission"
    ],
    "Resource": "arn:aws:ec2:{region}:{account_id}:network-interface/**",
    "Condition": {
    "StringLike": {
    "aws:ResourceTag/AMAZON_Q":
    "qbusiness_{account_id}_{application_id}*"
    }
    }
},
{
    "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
    "Effect": "Allow",
    "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeVpcs"
    ]
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
            "Effect": "Allow",
            "Principal": {
                "Service": "qbusiness.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "{{source_account}}"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
                }
            }
        }
    ]
}
```

For more information on Amazon Q data source connector IAM roles, see [IAM roles for Amazon Q data source connectors](#).

## Connecting Amazon RDS (Oracle) to Amazon Q

Amazon RDS (Oracle) (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. You can connect your Amazon RDS (Oracle) to Amazon Q by following these steps:

1. **Create a VPC**: Set up a Virtual Private Cloud (VPC) in your AWS account.
2. **Assign Proper Permissions**: Ensure that your VPC has the necessary permissions to interact with Amazon Q services.
3. **Set Up a Subnet**: Create a subnet within your VPC that is isolated and secure.
4. **Configure Security Groups**: Limit access to your database by configuring security groups that allow only the required network traffic.
5. **Connect to Amazon Q**: Use the trust policy mentioned above to allow Amazon Q to assume the role and connect to the RDS instance.

By following these steps, you can securely connect your Amazon RDS (Oracle) instance to Amazon Q, enabling real-time data integration and analytics.
RDS (Oracle) instance to Amazon Q – using either the AWS Management Console, CLI, or the CreateDataSource API – and create an Amazon Q web experience.

The Amazon Q Aurora (MySQL) data source connector supports Amazon RDS Oracle Database 21c, Oracle Database 19c, Oracle Database 12c.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Amazon RDS (Oracle), make sure you have:

- Noted your database user name and password.

  **Important**
  As a best practice, provide Amazon Q with read-only database credentials.

- Copied your database host URL, port, and instance.

In your AWS account, make sure you have:
• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your Amazon RDS (Oracle) authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Amazon RDS (Oracle) page, enter the following information:

1. **Name** – Name your data source for easy tracking.

   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Source**, enter the following information:
   a. **Host** – Enter the database host name.
   b. **Port** – Enter the database port.
   c. **Instance** – Enter the database instance.
   d. **SSL certificate location** – Choose to enter the Amazon S3 path to your SSL certificate file.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.

   Note
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).

4. In **Authentication** – Enter the following information for your AWS Secrets Manager secret.
a. **Secret name** – A name for your secret.

b. For **Database user name**, and **Password** – Enter the authentication credential values you copied from your database.

c. Choose **Save**.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

7. In **Sync scope**, enter the following information:

   - **SQL query** – Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Q 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 names of the document body column within your database table.

8. In **Additional configuration – optional** – Configure the following settings:

   - **Change-detecting columns** – Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q 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 Q 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.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   • **Full sync** – Sync all content regardless of the previous sync status.
   
   • **New or modified content sync** – Sync only new and modified documents.
   
   • **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

For more details, see **Sync mode**.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see **Sync run schedule**.

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See **Tags** for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

For more information, see **Field mappings**.
13. To finish connecting your data source to Amazon Q, select **Add data source**.

You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

**Note**

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

### Using the API

You use the **CreateDataSource** action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

### JSON schema

The following is the JSON schema:

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "connectionConfiguration": {
            "type": "object",
            "properties": {
                "repositoryEndpointMetadata": {
                    "type": "object",
                    "properties": {
                        "dbType": {
                            "type": "string",
                            "enum": [
                                "mysql",
                                Amazon RDS (Oracle)
                            ]
                        }
                    }
                }
            }
        }
    }
}
```
"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"
                }
              }
            }
          ]
        }
      }
    }
  }
}
"type": "string",
},
"dataSourceFieldName": {
  "type": "string"
},
"required": [
  "indexFieldName",
  "indexFieldType",
  "dataSourceFieldName"
]
}
],
"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"
},
"serverlessAurora": {
  "type": "string",
  "enum": ["true", "false"]
},
"required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
"type": {
  "type": "string",
  "pattern": "JDBC"
},
"syncMode": {
  "type": "string",
  "enum": ["FORCED_FULL_CRAWL", "FULL_CRAWL"
  ]
},
"secretArn": {
  "type": "string"
},
"version": {
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>Required configuration information for connecting your data source.</td>
</tr>
<tr>
<td></td>
<td>• dbType—The type of Java database you are using, whether mysql, db2, postgresql, oracle, or sqlserver.</td>
</tr>
<tr>
<td></td>
<td>• dbHost—The database host name.</td>
</tr>
<tr>
<td></td>
<td>• dbPort—The database port.</td>
</tr>
<tr>
<td></td>
<td>• dbInstance—The database instance.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>document</td>
<td>A list of objects that map the attributes or field names of your database content to Amazon Q index field names. For more information, see <a href="#">Mapping data source fields</a>.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.</td>
</tr>
<tr>
<td>primaryKey</td>
<td>Provide the primary key for the database table. This identifies a table within your database.</td>
</tr>
<tr>
<td>titleColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>bodyColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>sqlQuery</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>timestampColumn</td>
<td>Enter the name of the column which contains time stamps. Amazon Q uses time stamp information to detect changes in your content and sync only changed content.</td>
</tr>
<tr>
<td>timestampFormat</td>
<td>Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.</td>
</tr>
<tr>
<td>timezone</td>
<td>Enter the name of the column which contains time zones for the content to be crawled.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>changeDetectingColumns</td>
<td>Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q will re-index content when there is a change in any of these columns</td>
</tr>
<tr>
<td>allowedUsersColumns</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>allowedGroupsColumn</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>sourceURIColumn</td>
<td>Enter the name of the column which contains Source URLs to be indexed.</td>
</tr>
<tr>
<td>isSslEnabled</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify JDBC as your data source type.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose:</td>
</tr>
<tr>
<td></td>
<td>• FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index.</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:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;user name&quot;: &quot;database user name&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;password&quot;: &quot;password&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>version</td>
<td>The version of the template that is currently supported.</td>
</tr>
</tbody>
</table>

### ACL crawling

When you connect a database data source to Amazon Q, Amazon Q crawls user and group information from a column in the source table. You specify this column in the console or using the configuration parameter as part of the CreateDataSource operation.
If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- Authorization
- Identity crawler
- Understanding User Store

IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- Permission to access the SSL certificate stored in your Amazon S3 bucket.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowsAmazonQToGetS3Objects",
    "Action": [
      "s3:GetObject"
    ],
```

Amazon RDS (Oracle)
"Resource": [
    "arn:aws:s3:::{{input_bucket_name}}/*"
],
"Effect": "Allow",
"Condition": {
    "StringEquals": {
        "aws:ResourceAccount": "{{account_id}}"
    }
}
],
{
    "Sid": "AllowsAmazonQToGetSecret",
    "Effect": "Allow",
    "Action": [
        "secretsmanager:GetSecretValue"
    ],
    "Resource": [
        "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:{{secret_id}}"
    ]
},
{
    "Sid": "AllowsAmazonQToDecryptSecret",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:{{region}}:{{account_id}}:key:{{key_id}}"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
                "secretsmanager.*.amazonaws.com"
            ]
        }
    }
},
{
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ]
}
"Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}",

  ],  "Resource": [    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/*"
  ]}

{  "Sid": "AllowsAmazonQToCreateAndDeleteNI",  "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]]"
  ]}

{  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",  "Effect": "Allow",  "Action": [    "ec2:CreateNetworkInterface",    "ec2:DeleteNetworkInterface"
  ],  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",  "Condition": {  }  }
"StringLike": {
    "aws:RequestTag/AMAZON_Q":
    "qbusiness_{{account_id}}_{{application_id}}_***"  
    
},

    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "AMAZON_Q"
      ]
    }
  }
}
{
  "Sid": "AllowsAmazonQToCreateTags",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
    }
  }
},
{
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/AMAZON_Q":
      "qbusiness_{{account_id}}_{{application_id}}_***"
    }
  }
},
{
  "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeAvailabilityZones",

To allow Amazon Q to assume a role, you must also use the following trust policy:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
         "Effect": "Allow",
         "Principal": {
            "Service": "qbusiness.amazonaws.com"
         },
         "Action": "sts:AssumeRole",
         "Condition": {
            "StringEquals": {
               "aws:SourceAccount": "{{source_account}}"
            },
            "ArnLike": {
               "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
            }
         }
      }
   ]
}
```

For more information on Amazon Q data source connector IAM roles, see [IAM roles for Amazon Q data source connectors](#).
Connecting Amazon RDS (PostgreSQL) to Amazon Q

Amazon RDS (PostgreSQL) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. If you are an AWS user, you can use Amazon Q to index your Amazon RDS (PostgreSQL) data source.

The Amazon Q Amazon RDS (PostgreSQL) data source connector supports PostgreSQL 9.6.

You can connect your Amazon RDS (PostgreSQL) instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Amazon RDS (PostgreSQL), make sure you have:

- Noted your database user name and password.

⚠️ Important

As a best practice, provide Amazon Q with read-only database credentials.
• Copied your database host URL, port, and instance. You can find this information on the Amazon RDS console.

In your AWS account, make sure you have:

• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your Amazon RDS (PostgreSQL) authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

**Note**
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

**Using the console**

On the Amazon RDS (PostgreSQL) page, enter the following information:

1. **Name** – Name your data source for easy tracking.

   **Note**: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Source**, enter the following information:
   a. **Host** – Enter the database host URL, for example: `http://instance URL.region.rds.amazonaws.com`.
   b. **Port** – Enter the database port, for example, 5432.
   c. **Instance** – Enter the database instance, for example `postgres`.
   d. **SSL certificate location** – Choose to enter the Amazon S3 path to your SSL certificate file.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.
Observation
Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).

4. In **Authentication** – Enter the following information for your **AWS Secrets Manager secret**.

   a. **Secret name** – A name for your secret.
   
   b. For **Database user name**, and **Password** – Enter the authentication credential values you copied from your database.
   
   c. Choose **Save**.

5. **Configure VPC and security group** – *optional* – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

7. In **Sync scope**, enter the following information:

   - **SQL query** – Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Q 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.
8. In Additional configuration – optional – Configure the following settings:

- **Body column** – Provide the names of the document body column within your database table.

- **Change-detecting columns** – Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q 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 Q 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.

9. In Sync mode, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

- **Full sync** – Sync all content regardless of the previous sync status.

- **New or modified content sync** – Sync only new and modified documents.

- **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

For more details, see **Sync mode**.

10. In Sync run schedule, for Frequency – Choose how often Amazon Q will sync with your data source. For more details, see **Sync run schedule**.

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See **Tags** for more details.
12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.
   
   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

   **Note**
   You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

### Using the API

You use the **CreateDataSource** action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

### JSON schema

The following is the JSON schema:

```json
{
    "$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"
},
"serverlessAurora": {
    "type": "string",
    "enum": ["true", "false"]
},
"required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
},
"type": {
    "type": "string",
    "pattern": "JDBC"
"syncMode": {
    "type": "string",
    "enum": [
        "FORCED_FULL_CRAWL",
        "FULL_CRAWL"
    ]
},
"secretArn": {
    "type": "string"
},
"version": {
    "type": "string",
    "anyOf": [
        {
            "pattern": "1.0.0"
        }
    ]
},
"required": [
    "connectionConfiguration",
    "repositoryConfigurations",
    "syncMode",
    "additionalProperties",
    "secretArn",
    "type"
]
}

The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>Required configuration information for connecting your data source.</td>
</tr>
<tr>
<td>dbType—The type of Java database you are using, whether mysql, db2, postgresql, oracle, or sqlserver.</td>
<td></td>
</tr>
</tbody>
</table>

Amazon RDS (PostgreSQL)
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbHost</td>
<td>The database host name.</td>
</tr>
<tr>
<td>dbPort</td>
<td>The database port.</td>
</tr>
<tr>
<td>dbInstance</td>
<td>The database instance.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>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.</td>
</tr>
<tr>
<td>document</td>
<td>A list of objects that map the attributes or field names of your database content to Amazon Q index field names. For more information, see <a href="#">Mapping data source fields</a>.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.</td>
</tr>
<tr>
<td>primaryKey</td>
<td>Provide the primary key for the database table. This identifies a table within your database.</td>
</tr>
<tr>
<td>titleColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>bodyColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>sqlQuery</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>timestampColumn</td>
<td>Enter the name of the column which contains time stamps. Amazon Q uses time stamp information to detect changes in your content and sync only changed content.</td>
</tr>
<tr>
<td>timestampFormat</td>
<td>Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.</td>
</tr>
<tr>
<td>timezone</td>
<td>Enter the name of the column which contains time zones for the content to be crawled.</td>
</tr>
<tr>
<td>changeDetectingColumns</td>
<td>Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q will re-index content when there is a change in any of these columns.</td>
</tr>
<tr>
<td>allowedUsersColumns</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>allowedGroupsColumn</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>sourceURIColumn</td>
<td>Enter the name of the column which contains Source URLs to be indexed.</td>
</tr>
<tr>
<td>isSslEnabled</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify JDBC as your data source type.</td>
</tr>
</tbody>
</table>
## Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
</table>
| syncMode      | Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose  
  - **FORCED_FULL_CRAWL** to freshly re-crawl all content and replace existing content each time your data source syncs with your index  
  - **FULL_CRAWL** to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index  
  - **CHANGE_LOG** to incrementally crawl only new and modified content each time your data source syncs with your index. |
| 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:  
  ```json  
  {  
    "user name": "database user name",  
    "password": "password"  
  }  
  ``` |
| version       | The version of the template that is currently supported. |

### ACL crawling

When you connect a database data source to Amazon Q, Amazon Q crawls user and group information from a column in the source table. You specify this column in the console or using the configuration parameter as part of the CreateDataSource operation.
If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- Authorization
- Identity crawler
- Understanding User Store

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- Permission to access the SSL certificate stored in your Amazon S3 bucket.
- (Optional) If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowsAmazonQToGetS3Objects",
         "Action": [
            "s3:GetObject"
         ],
```
"Resource": [
    "arn:aws:s3:::{{input_bucket_name}}/*"
],
"Effect": "Allow",
"Condition": {
    "StringEquals": {
        "aws:ResourceAccount": "{{account_id}}"
    }
}
],
{
    "Sid": "AllowsAmazonQToGetSecret",
    "Effect": "Allow",
    "Action": [
        "secretsmanager:GetSecretValue"
    ],
    "Resource": [
        "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:{{secret_id}}"
    ]
},
{
    "Sid": "AllowsAmazonQToDecryptSecret",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:{{region}}:{{account_id}}:key:{{key_id}}"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": ["secretsmanager.*.amazonaws.com"
        ]
    }
},
{
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ],
}
"Resource": "arn:aws:qbusiness:{region}:{source_account}:application/{{application_id}}/index/{{index_id}}",
},
{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroups"
  ],
  "Resource": [
    "arn:aws:qbusiness:{region}:{account_id}:application/{{application_id}}",
    "arn:aws:qbusiness:{region}:{account_id}:application/{{application_id}}/index/{{index_id}}",
    "arn:aws:qbusiness:{region}:{account_id}:application/{{application_id}}/index/{{index_id}}/data-source/**"
  ]
},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "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]]"
  ]
},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNIFormsModuleTag",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:{region}:{account_id}:network-interface/**",
  "Condition": {
    "StringLike": {
      "ec2:NetworkInterfaceOwnerId": {"AWS::AccountId": "[[account_id]]"}
    }
  }
}
"StringLike": {
   "aws:RequestTag/AMAZON_Q":
   "qbusiness_{{account_id}}_{{application_id}}_**"
 },
   "ForAllValues:StringEquals": {
     "aws:TagKeys": [
       "AMAZON_Q"
     ]
   }
 }
},
{
   "Sid": "AllowsAmazonQToCreateTags",
   "Effect": "Allow",
   "Action": [
     "ec2:CreateTags"
   ],
   "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
   "Condition": {
     "StringEquals": {
       "ec2:CreateAction": "CreateNetworkInterface"
     }
   }
 },
{
   "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
   "Effect": "Allow",
   "Action": [
     "ec2:CreateNetworkInterfacePermission"
   ],
   "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
   "Condition": {
     "StringLike": {
       "aws:ResourceTag/AMAZON_Q":
       "qbusiness_{{account_id}}_{{application_id}}_**"
     }
   }
 },
{
   "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
   "Effect": "Allow",
   "Action": [
     "ec2:DescribeNetworkInterfaces",
     "ec2:DescribeAvailabilityZones",
     "ec2:DescribeNetworkInterfacePermissions"
   ]
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
        }
      }
    }
  ]
}
```

For more information on Amazon Q data source connector IAM roles, see [IAM roles for Amazon Q data source connectors](#).
Connecting Amazon S3 to Amazon Q

Amazon Simple Storage Service (Amazon S3) is an object storage service that stores data as objects within storage buckets. You can connect an Amazon S3 instance to Amazon Q—using either the AWS Management Console or the `CreateDataSource` API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- Adding document metadata
- Adding ACL information using metadata file
- IAM role for Amazon S3 data sources

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Amazon S3, make sure you have:

- Copied the name of your Amazon S3 bucket name.

Note

Your bucket must be in the same AWS Region as your Amazon Q index, and your index must have permissions to access the bucket that contains your documents.

In your AWS account, make sure you have:
• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your Amazon S3 authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Amazon S3 page, enter the following information:

1. Name – Name your data source for easy tracking.

   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. Configure VPC and security group – optional – You can choose to use a VPC if your Amazon S3 bucket is not accessible through the public internet. If you so, you must add Subnets and VPC security groups as well.

   Important
   Make sure you have:
   • Configured your VPC according to the steps in Gateway endpoints for Amazon S3.
   • Chosen a private subnet in an Amazon Q supported availability zone.
   • Configured your security group to allow Amazon Q to access the Amazon S3 endpoint.

   If you choose to use VPC, enter the following information:
a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

For more information, see [Using Amazon VPC](#).

3. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   - **Note**
     
     IAM roles used for applications can't be used for data sources. If you are unsure if an existing role is used for an application, choose **Create a new role** to avoid errors.

4. **Sync scope**, enter the following information:

   a. **Enter the data source location** – The path to the Amazon S3 bucket where your data is stored. Select **Browse S3** to find and choose your bucket.

   b. **Maximum file size** - *optional* – The maximum file size value that Amazon Q will crawl. Amazon Q will only crawl files within the limit you define.

   c. **Advanced settings**, enter the following information:

      - **Metadata files prefix folder location** - *optional* – The path to the folder in which your metadata is stored. Select **Browse S3** to locate your metadata folder.

      - **Access control list configuration file location** - *optional* – The path to the location of a file containing a JSON structure that specifies access settings for the files stored in your S3 data source. Select **Browse S3** to locate your ACL file.

   d. **Regex patterns** – Add patterns to include or exclude documents from your index. All paths are relative to the data source location Amazon S3 bucket. You can add up to 100 patterns.

      You can include and exclude documents using file names, file types, file paths, and glob patterns (patterns that can expand a wildcard pattern into a list of path names that match the given pattern).
Examples of glob patterns include:

- `/myapp/config/*` – All files inside config directory
- `/**/*.png` – All .png files in all directories
- `/**/*.{png,ico,md}` – All .png, .ico, or .md files in all directories
- `/myapp/src/**/*.{ts}` – All .ts files inside src directory (and all its subdirectories)
- `/**/!(*.module).ts` – All .ts files but not .module.ts

5. **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

6. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

7. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

8. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

9. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

10. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.
Note

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

Using the API

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

Amazon S3 JSON schema

The following is the Amazon S3 JSON schema:

```json
{
  "$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"
                }
              }
            }
          ]
        }
      }
    }
  }
},
"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"
},
"maxFileSizeInMegaBytes": {
"type": "string"
}
}
},
"syncMode": {
"type": "string",
"enum": [
"FULL_CRAWL",
"FORCED_FULL_CRAWL"
]
},
"type": {
"type": "string",
"pattern": "S3"
},
"version": {
"type": "string",
"anyOf": [
{
"pattern": "1.0.0"
}
]
}
The following provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>BucketName</td>
<td>The name of your Amazon S3 bucket.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source</td>
</tr>
<tr>
<td>• inclusionPatterns</td>
<td>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 file matches both an inclusion and exclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.</td>
</tr>
<tr>
<td>• exclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionPrefixes</td>
<td></td>
</tr>
<tr>
<td>• exclusionPrefixes</td>
<td></td>
</tr>
<tr>
<td>aclConfigurationFilePath</td>
<td>The path to the file that controls access control information for your documents in an Amazon Q index.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>metadataFilesPrefix</td>
<td>The location, in your Amazon S3 bucket, of your document metadata files.</td>
</tr>
<tr>
<td>maxFileSizeInMegaBytes</td>
<td>The maximum size, in megabytes, of a file that can be added to your Amazon Q index.</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose from the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify S3 as your data source type.</td>
</tr>
<tr>
<td>version</td>
<td>The version of the template that's supported.</td>
</tr>
</tbody>
</table>

### Adding document metadata

To customize chat results for your end users, you can add metadata to documents in an Amazon S3 bucket by using a metadata file. Metadata is additional information about a document, such as its title and the date and time it was created.

⚠️ **Note**

For more information about how document attributes can help you customize chat results for your end users, see ???.

---

Amazon S3

319
Each metadata file is associated with an indexed document. Your metadata files must be stored in the same S3 bucket as your indexed files. You can specify a location within the S3 bucket for your metadata files by using the AWS Management Console. Or, you can use the metadataFilesPrefix field of the Amazon S3 configuration parameter using the JSON schema 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 Q 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. 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, the metadata's Amazon S3 prefix, and the .metadata.json suffix must be no more than a total of 1,024 characters. We recommend that your Amazon S3 key is less than 1,000 characters to account for additional characters when combining your key with the prefix and suffix.

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
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 uses the following template.

```json
{
    "DocumentId": "document ID",
    "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"
}
```

All of the attributes and fields are optional, so it's not necessary to include all attributes. However, you must provide a value for each attribute that you want to include; the value can't be empty. If you don't specify the `_source_uri`, the links returned by Amazon Q in the chat results point to the Amazon S3 bucket that contains the document.

**Note**

For information about supported document types, see [Supported document types](#).
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.

You can use the AccessControlList field to filter the response from a query. This way, only certain users and groups have access to documents.

**Adding ACL information using metadata file**

You add access control information to a document in an Amazon S3 data source using a metadata file associated with the document. You specify the file using the console or as the aclConfigurationFilePath parameter when you call the CreateDataSource or UpdateDataSource API and use the configuration parameter.

The configuration file contains a JSON structure that identifies an Amazon 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.

You provide three pieces of information in the file:

- The access that the entity should have. You can use ALLOW or DENY.
- The type of entity. You can use USER or GROUP.
- The name of the entity.

The JSON structure for the configuration file must be in the following format:

```json
[
    {
        "keyPrefix": "s3://BUCKETNAME/prefix1/",
        "aclEntries": [
            {
                "Name": "user1",
                "Type": "USER",
                "Access": "ALLOW"
            },
            {
                "Name": "group1",
                "Type": "GROUP",
                "Access": "ALLOW"
            }
        ]
    }
]
```
"Type": "GROUP",
   "Access": "DENY"
  
]}
],
{
   "keyPrefix": "s3://BUCKETNAME/prefix2/",
   "aclEntries": [
      {
         "Name": "user2",
         "Type": "USER",
         "Access": "ALLOW"
      },
      {
         "Name": "user1",
         "Type": "USER",
         "Access": "DENY"
      },
      {
         "Name": "group1",
         "Type": "GROUP",
         "Access": "DENY"
      }
   ]
}

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

**IAM role for Amazon S3 data sources**

When you use an Amazon S3 bucket as a data source, you must provide a role that has permissions to:

- Access your Amazon S3 bucket.
- Permission to access the [BatchPutDocument](#) and [BatchDeleteDocument](#) API operations in order to ingest documents.
• Permission to access the Principal Store APIs needed to ingest access control and identity information from documents.

**Note**

Amazon Q doesn't use a bucket policy that grants permissions to an Amazon Q principal to interact with an Amazon S3 bucket. Instead, Amazon Q uses IAM roles. To avoid any data security issues in accidentally granting permissions to arbitrary principals, make sure that Amazon Q isn't included as a trusted member in your bucket policy. However, you can add a bucket policy to use an Amazon S3 bucket across different AWS accounts.

To allow Amazon Q to use an Amazon S3 bucket as a data source, use the following role policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToGetObjectfromS3",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::{{input_bucket_name}}/*"
            ],
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "{{account_id}}"
                }
            }
        },
        {
            "Sid": "AllowsAmazonQToListS3Buckets",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::{{input_bucket_name}}"
            ],
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "{{account_id}}"
                }
            }
        }
    ]
}
```
If the documents in the Amazon S3 bucket are encrypted, you must provide the following permissions to use the AWS KMS key to decrypt the documents:

```json
{
    "Sid": "AllowsAmazonQToDecryptSecret",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ],
    "Resource": ["arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",

    "Sid": "AllowsAmazonQToCallPrincipalMappingAPIs",
    "Effect": "Allow",
    "Action": ["qbusiness:PutGroup",
                "qbusiness:CreateUser",
                "qbusiness:DeleteGroup",
                "qbusiness:UpdateUser",
                "qbusiness:ListGroups"
            ],
    "Resource": ["arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
                "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
                "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/"
            ]
}
```
"Action": [
    "kms:Decrypt"
],
"Resource": [
    "arn:aws:kms:{region}:{{account_id}}:key/{[key_id]}"
],
"Condition": {
    "StringLike": {
        "kms:ViaService": [
            "secretsmanager.*.amazonaws.com"
        ]
    }
}
}

If you are using an Amazon VPC, you must add the following VPC access permissions to your policy:

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToGetObjectfromS3",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::{{input_bucket_name}}/*"
            ],
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "{{account_id}}"
                }
            }
        },
        {
            "Sid": "AllowsAmazonQToListS3Buckets",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::{{input_bucket_name}}"
            ]
        }
    ]
}
"Effect": "Allow",
"Condition": {
    "StringEquals": {
        "aws:ResourceAccount": "{{account_id}}"
    }
},
",
"Sid": "AllowsAmazonQToIngestDocuments",
"Effect": "Allow",
"Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
],
"Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/
{{application_id}}/index/{{index_id}}"
},
",
"Sid": "AllowsAmazonQToCallPrincipalMappingAPIs",
"Effect": "Allow",
"Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroups"
],
"Resource": [
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/"
]}
,
"Sid": "AllowsAmazonQToCreateAndDeleteENI",
"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]]"
],
"Sid": "AllowsAmazonQToCreateDeleteENI",
"Effect": "Allow",
"Action": [
  "ec2:CreateNetworkInterface",
  "ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{region}::{account_id}::network-interface/**",
"Condition": {
  "StringLike": {
    "aws:RequestTag/AMAZON_Q": "qbusiness_{account_id}_{application_id}_*"
  },
  "ForAllValues:StringEquals": {
    "aws:TagKeys": [
      "AMAZON_Q"
    ]
  }
}
],
"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
  "ec2:CreateTags"
],
"Resource": "arn:aws:ec2:{region}::{account_id}::network-interface/**",
"Condition": {
  "StringEquals": {
    "ec2:CreateAction": "CreateNetworkInterface"
  }
}
],
"Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
"Effect": "Allow",
"Action": [
  "ec2:CreateNetworkInterfacePermission"
],
"Resource": "arn:aws:ec2:{region}::{account_id}::network-interface/**",
"Condition": {
To allow Amazon Q to assume a role, use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
        }
      }
    }
  ]
}
```
Connecting Amazon Q custom connector to Amazon Q

Use a custom data source when you have a repository that Amazon Q doesn't yet provide a data source connector for. When you create a custom data source, you have complete control over how the documents to index are selected. Amazon Q 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 can use a custom data source connector to:

• See the same run history metrics that Amazon Q data sources provide even when you can't use Amazon Q data sources to sync your repositories.

• Create a consistent sync monitoring experience between Amazon Q data sources and custom data sources.

• See sync metrics for a data source connector that you created using the `BatchPutDocument` and `BatchDeleteDocument` API operations.

You can create an Amazon Q custom data source connector using either the AWS Management Console or the `CreateDataSource`.

When you create a custom data source using the `CreateDataSource` API operation:

• The action returns an ID to use when you synchronize the data source.

• You can't set the `Configuration`, `RoleArn`, or `Schedule` parameters. If you set these parameters, Amazon Q returns a `ValidationException` exception.

• You must specify the main title of your documents using the `Document` object, and `_source_uri` in `DocumentAttribute`. The main title is required so that `DocumentTitle` and `DocumentURI` are included in the `ChatSync` response.

When you create a custom data source using the console:

• The console returns an ID to use when you synchronize the data source.

• 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.

Topics
• Creating an Amazon Q custom connector
• Required attributes
• Viewing metrics

Creating an Amazon Q custom connector

To use a custom data source, create an application that is responsible for updating your Amazon Q index. The application depends on a crawler that you create. The crawler reads the documents in your repository and determines which documents should be sent to Amazon Q. 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 operation to signal that a sync job is starting. You provide a data source ID to identify the data source that is synchronizing. Amazon Q returns an execution ID to identify a particular sync job.

   i Note
   After you end a sync job, you can start a new sync job. There can be a period of time before all of the submitted documents are added to the index. To see the status of the sync job, use the ListDataSourceSyncJobs operation. 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, SUCCEEDED, or SYNCING_INDEX.

3. To remove documents from the index, use the BatchDeleteDocument operation. 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. To signal the end of the sync job, use the StopDataSourceSyncJob operation. After you call the StopDataSourceSyncJob operation, the associated execution ID is no longer valid.
After you call the StopDataSourceSyncJob operation, you can't use a sync job identifier in a call to the BatchPutDocument or BatchDeleteDocument operations. If you do, all of the documents submitted are returned in the FailedDocuments response message from the API.

5. To list the sync jobs for the data source and to see metrics for the sync jobs, use the ListDataSourceSyncJobs operation with the index and data source identifiers.

**Required attributes**

When you submit a document to Amazon Q using the BatchPutDocument API operation, you must provide the following two attributes for each document:

- `_data_source_id` – The identifier of the data source. This is returned when you create the data source with either the console or the CreateDataSource API operation.
- `_data_source_sync_job_execution_id` – The identifier of the sync run. This is returned when you start the index synchronization with the StartDataSourceSyncJob operation.

The following is the JSON required to index a document using a custom data source.

```json
{
  "Documents": [
    {
      "Attributes": [
        {
          "Key": "_data_source_id",
          "Value": {
            "StringValue": "data source identifier"
          }
        },
        {
          "Key": "_data_source_sync_job_execution_id",
          "Value": {
            "StringValue": "sync job identifier"
          }
        }
      ]
    }
  ]
}
```
When you remove a document from the index using the BatchDeleteDocument API operation, you must 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 either the console or the CreateDataSource API operation.
- **DataSourceSyncJobId** – The identifier of the sync run. This is returned when you start the index synchronization with the StartDataSourceSyncJob operation.

The following is the JSON required to delete a document from the index using the BatchDeleteDocument operation.

```
{
  "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 operation to get the metrics associated with the sync job. Use this API operation to monitor your custom data source syncs.

You can submit the same document multiple times, either as part of the BatchPutDocument operation, the BatchDeleteDocument operation, or if the document is submitted for both...
addition and deletion, Regardless of how you submit the document, it is only counted once in the metrics.

- **DocumentsAdded** – The number of documents submitted using the BatchPutDocument operation associated with this sync job that are 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 operation associated with this sync job that are 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 documents were accepted by Amazon Q for indexing but could not be indexed or deleted. If a document isn't accepted by Amazon Q, the identifier for the document is returned in the FailedDocuments response property of the BatchPutDocument and BatchDeleteDocument operations.

- **DocumentsModified** – The number of modified documents submitted using the BatchPutDocument operation associated with this sync job that were modified in the Amazon Q index.

Amazon Q also emits Amazon CloudWatch metrics while indexing documents. For more information, see [Monitoring Amazon Q with Amazon CloudWatch](https://docs.aws.amazon.com/q/latest/dg/monitoring-cloudwatch.html).

Amazon Q doesn't return the **DocumentsScanned** metric for custom data sources.

**Connecting Amazon Q Web Crawler to Amazon Q**

An Amazon Q Web Crawler connector crawls and indexes either public facing websites or internal company websites that use HTTPS. With Amazon Q web crawler, you can create a generative AI web experience for your end users based on the website data you crawl using either the AWS Management Console or the [CreateDataSource](https://docs.aws.amazon.com/q/latest/dg/api acción=CreaDataSource) API.

Amazon Q Web Crawler uses the Selenium web crawler package and a Chromium driver. Amazon Q automatically updates the version of Selenium and the Chromium driver using continuous integration (CI).
Important

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 Q Web Crawler to index your own webpages, or webpages that you have authorization to index. To learn how to stop Amazon Q Web Crawler from indexing your websites, see Configuring the robots.txt file.

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.

Note

Amazon Q Web Crawler connector does not support AWS KMS encrypted Amazon S3 buckets. It supports only server-side encryption with Amazon S3 managed keys.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- IAM roles
- Configuring the robots.txt file
Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

For Amazon Q Web Crawler, make sure you have:

- Copied the seed or sitemap URLs of the websites that you want to index and stored them in a text file or an Amazon S3 bucket. Each URL must be included on a separate line.

- **For XML sitemaps:** Copied the sitemap XML and saved it in an XML file in an Amazon S3 bucket. You can also combine multiple sitemap XML files into a .zip file.

- **For websites that require basic, NTLM, or Kerberos authentication:**
  - Noted your website authentication credentials, which include a username and password.

  **Note**
  Amazon Q Web Crawler 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 username and password.
  - Copied the XPaths (XML Path Language) of the username field (and the username 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 follow this format: `//tagname[@Attribute='Value']`.

  **Note**
  Amazon Q Web Crawler 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 that you want to crawl. The web proxy must be public facing. Amazon Q supports connecting to web proxy servers 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 [Using Amazon VPC](#).
In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the API, noted the ARN of the IAM role.

- **For websites that require authentication credentials to crawl:** Stored your Web Crawler authentication credentials in an AWS Secrets Manager secret and, if using the API, noted the ARN of the secret.

**Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

**Using the console**

On the **Web Crawler** page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In **Source** choose from the following options:
   
   - **Source URLs** – Add up to 10 seed/starting point URLs of the websites you want to crawl. You can also include website subdomains.
   
   - **Source sitemaps** – Add up to 3 sitemap URLs of the websites you want to crawl.
   
   - **Source URLs file** – Add up to 100 seed/starting point URLs listed in a text file in Amazon S3. Each URL should be on a separate line in the text file.
   
   - **Source sitemaps file** – Add up to 3 sitemap XML files stored in Amazon S3. You can also zip the XML files.

**Note**

If you choose to use a text file that includes a list of up to 100 seed URLs or to use a sitemap XML file, you specify the path to the Amazon S3 bucket where your file is stored.
You can also combine 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/".

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 Q configures the data source using the seed URLs endpoint information in the Secrets Manager secret for authentication. Therefore, Amazon Q can't reconfigure the data source when changing to sitemaps.

3. In Authentication, choose the type of authentication you want to use and enter the following information in your AWS Secrets Manager secret:

- **No authentication** – Choose to crawl a public website without any authentication.
- **Basic authentication** – Enter a name for the secret, plus the username and password
- **NTLM/Kerberos authentication** – Enter a name for the secret, plus the username and password. NTLM authentication protocol includes password hashing, and Kerberos authentication protocol includes password encryption
- **Form authentication** – Enter a name for the secret, and the username and password. Use XPath for the username field. 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']
- **SAML authentication** – Enter a name for the secret, plus the username and password. Use XPath for the username field and for the username button. 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']

4. **Web proxy – optional** – Enter the host name and the port number of the proxy server that 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 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 secret that stores the credentials.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:
   
   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

7. In **Sync scope**, enter the following information:
   
   a. **Sync domain range** – Choose whether to sync website domains with subdomains only, or also crawl other domains that the webpages link to (**Sync everything**). By default, Amazon Q only syncs the domains of the websites that you want to crawl.
   
   b. In **Additional configuration – optional** – Configure the following settings:
      
      - **Scope settings**, choose from the following:
        
        - **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 webpage or attachment to crawl.
        
        - **Maximum links per page** – The maximum number of URLs on a single webpage to crawl.
• **Maximum throttling** – The maximum number of URLs crawled per website host per minute.

• **Include files that web pages link to** – Choose to crawl files that the webpages link to.

• **Crawl URL patterns** – Add regular expression patterns to include or exclude crawling specific URLs, and indexing any hyperlinks on these URL webpages.

• **URL pattern to index** – Add regular expression patterns to include or exclude crawling specific URLs, and indexing any hyperlinks on these URL webpages.

8. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   • **Full sync** – Sync all content regardless of the previous sync status.

   • **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

9. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see **Sync run schedule**.

10. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See **Tags** for more details.

11. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

    a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

    b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

    For more information, see **Field mappings**.

12. To finish connecting your data source to Amazon Q, select **Add data source**.

    You are taken to the **Data source details**, where you can view your data source configuration details.

13. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.
Note

You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

Using the API

To connect Amazon Q to application-name using the Amazon Q API, call CreateDataSource. Use this API to:

- provide a name and tags for your data source
- an Amazon Resource Name (ARN) of an IAM role with permission to access the data source and required resources
- a sync schedule for Amazon Q to check the documents in your data source
- a Amazon VPC configuration

For more information on available parameters, see CreateDataSource in the Amazon Q API reference.

Provide the seed or starting point URLs, or the sitemap URLs, as part of the connection configuration or repository endpoint details. Also specify the website authentication credentials and authentication type if your websites require authentication, and other necessary configurations.

Web Crawler JSON schema

The following is the Web Crawler JSON schema:

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "connectionConfiguration": {
            "type": "object",
            "properties": {
                "connectionConfiguration": {
                    "type": "object",
                    "properties": {
```

```json
```
"repositoryEndpointMetadata": {
  "type": "object",
  "properties": {
    "siteMapUrls": {
      "type": "array",
      "items": {
        "type": "string",
        "pattern": "https://.*"
      }
    },
    "s3SeedUrl": {
      "type": ["string", "null"],
      "pattern": "s3:.*"
    },
    "s3SiteMapUrl": {
      "type": ["string", "null"],
      "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"
]
"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
}
},
"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",
      "connectionCredentials"
   ]
}
The following provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>siteMapUrls</td>
<td>The list of sitemap URLs for the websites that you want to crawl. You can list up to three sitemap URLs.</td>
</tr>
<tr>
<td>s3SeedUrl</td>
<td>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.</td>
</tr>
<tr>
<td>s3SiteMapUrl</td>
<td>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 file and store the .zip file in your Amazon S3 bucket.</td>
</tr>
<tr>
<td>seedUrlConnections</td>
<td>The list of seed or starting point URLs for the websites that you want to crawl. You can list up to 100 seed URLs.</td>
</tr>
<tr>
<td>seedUrl</td>
<td>The seed or starting point URL.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>authentication</td>
<td>The authentication type if your websites require the same authentication, otherwise specify NoAuthentication.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• webpage</td>
<td>A list of objects that map the attributes or field names of your webpages and webpage files to Amazon Q index field names. For example, the HTML webpage title tag can be mapped to the _document_title index field.</td>
</tr>
<tr>
<td>• attachment</td>
<td></td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>rateLimit</td>
<td>The maximum number of URLs crawled per website host per minute.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>maxFileSize</td>
<td>The maximum size (in MB) of a webpage or attachment to crawl.</td>
</tr>
<tr>
<td>crawlDepth</td>
<td>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.</td>
</tr>
<tr>
<td>maxLinksPerUrl</td>
<td>The maximum number of URLs on a webpage to include when crawling a website. This number is per webpage. As a website's webpages are crawled, any URLs that the webpages link to also are crawled. URLs on a webpage are crawled in order of appearance.</td>
</tr>
<tr>
<td>crawlSubDomain</td>
<td>true to crawl the website domains with subdomains only. For example, if the seed URL is &quot;abc.example.com&quot;, then &quot;a.abc.example.com&quot; and &quot;b.abc.example.com&quot; are also crawled. If you don't set crawlSubDomain or crawlAllDomain to true, then Amazon Q only crawls the domains of the websites that you want to crawl.</td>
</tr>
<tr>
<td>crawlAllDomain</td>
<td>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 Q only crawls the domains of the websites that you want to crawl.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>honorRobots</td>
<td>Set to true to respect the robots.txt directives of the websites that you want to crawl. These directives control how Amazon Q Web Crawler crawls the websites, and whether Amazon Q can crawl only specific content or not crawl any content.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
</tr>
<tr>
<td></td>
<td>The honorRobots feature is currently only available if you use the API.</td>
</tr>
<tr>
<td>crawlAttachments</td>
<td>Set to true to crawl files that the webpages link to.</td>
</tr>
<tr>
<td>inclusionURLCrawlPatterns</td>
<td>A list of regular expression patterns to include crawling certain URLs and indexing any hyperlinks on these URL webpages. 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 and website's webpages aren't included in the index.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• exclusionURLCrawlPatterns</td>
<td>A list of regular expression patterns to <em>exclude</em> crawling certain URLs and indexing any hyperlinks on these URL webpages. 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 webpages aren't included in the index.</td>
</tr>
<tr>
<td>• exclusionURLIndexPatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionFileIndexPatterns</td>
<td>A list of regular expression patterns to <em>include</em> 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.</td>
</tr>
<tr>
<td>exclusionFileIndexPatterns</td>
<td>A list of regular expression patterns to <em>exclude</em> certain webpage 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.</td>
</tr>
<tr>
<td>proxy</td>
<td>Configuration information required to connect to your internal websites through a web proxy.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>host</td>
<td>The host name of the proxy server that you want to use to connect to internal websites. For example, the host name of <a href="https://a.example.com/page1.html">https://a.example.com/page1.html</a> is &quot;a.example.com&quot;.</td>
</tr>
<tr>
<td>port</td>
<td>The port number of the proxy server that you want to use to connect to internal websites. For example, 443 is the standard port for HTTPS.</td>
</tr>
<tr>
<td>secretArn (proxy)</td>
<td>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.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify WEBCRAWLER RV2 as your data source type.</td>
</tr>
</tbody>
</table>

Amazon Q Web Crawler
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secretARN</td>
<td>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 NTLM/Kerberos, enter the username and password. The JSON keys in the secret must be <code>userName</code> and <code>password</code>. NTLM authentication protocol includes password hashing, and Kerberos authentication protocol includes password encryption. If you use SAML or form authentication, enter the username and password, XPath for the username field (and username button if using SAML), XPaths for the password field and button, and the login page URL. The JSON keys in the secret must be <code>userName</code>, <code>password</code>, <code>userNameFieldXpath</code>, <code>userNameButtonXpath</code>, <code>passwordFieldXpath</code>, <code>passwordButtonXpath</code>, and <code>loginPageUrl</code>. You can find the XPaths (XML Path Language) of elements using your web browser’s developer tools. XPaths usually follow this format: <code>//tagName[@Attribute='Value']</code>. Amazon Q 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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>

**IAM roles**

To connect Amazon Q Web Crawler to Amazon Q, you must give Amazon Q an IAM role that has the following permissions.

**If you're crawling a public website with no authentication:**

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store operations to ingest access control information from documents.

```json
{
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ],
    "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},
{
    "Sid": "AllowsAmazonQToIngestPrincipalMapping",
    "Effect": "Allow",
    "Action": [
        "qbusiness:PutGroup",
        "qbusiness:CreateUser",
        "qbusiness:DeleteGroup",
        "qbusiness:UpdateUser",
        "qbusiness:ListGroup"
    ],
    "Resource": [
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}"
    ]
}
```
If you're crawling a website which uses authentication:

- 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.

```json
{
    "Sid": "AllowsAmazonQToGetSecret",
    "Effect": "Allow",
    "Action": [
        "secretsmanager:GetSecretValue"
    ],
    "Resource": [
        "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
    ]
}
```

If your Secrets Manager secret is decrypted, add permissions for a AWS KMS key to decrypt the user name and password secret stored by Secrets Manager:

```json
{
    "Sid": "AllowsAmazonQToDecryptSecret",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:{{region}}:{{account_id}}:key:[[key_id]]"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
                "secretsmanager.*.amazonaws.com",
                "secretsmanager.*.amazonaws.com"
            ]
        }
    }
}
```
If your Amazon Q data source connector needs access to an object stored in an Amazon S3 bucket—like seed URLs or sitemaps—you must add the following permissions to your IAM role:

```json
{
    "Sid": "AllowsAmazonQToGetS3Objects",
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::{{input_bucket_name}}/*"
    ],
    "Effect": "Allow",
    "Condition": {
        "StringEquals": {
            "aws:ResourceAccount": "{{account_id}}"
        }
    }
}
```

If you are using an Amazon VPC, you need to add the following VPC access permissions to your policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowsAmazonQToCreateAndDeleteNI",
        "Effect": "Allow",
        "Action": [
            "ec2:CreateNetworkInterface",
            "ec2:DeleteNetworkInterface"
        ],
        "Resource": [
```

---

**Note**

Check that the file path to the object in your Amazon S3 bucket is of the following format:

`s3://BucketName/FolderName/FileName.extension`. 
"arn:aws:ec2:{{region}}:{{account_id}}:subnet/[[subnet_ids]]",
"arn:aws:ec2:{{region}}:{{account_id}}:security-group/[[security_group]]"
],
{
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterface",
"ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringLike": {
"aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
},
"ForAllValues:StringEquals": {
"aws:TagKeys": [
"AMAZON_Q"
]
}
}
},
{
"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
"ec2:CreateTags"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringEquals": {
"ec2:CreateAction": "CreateNetworkInterface"
}
}
},
{
"Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterfacePermission"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {

To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
        }
      }
    }
  ]
}
```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

**Configuring the robots.txt file**

Amazon Q Web Crawler respects standard robots.txt directives like Allow and Disallow. You can modify the robot.txt file of your website to control how Amazon Q Web Crawler crawls your website.

**Topics**

- Configuring how Amazon Q Web Crawler accesses your website
- Stopping Amazon Q Web Crawler from crawling your website

**Configuring how Amazon Q Web Crawler accesses your website**

You can control how the Amazon Q 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 Q Web Crawler to crawl all web pages except disallowed web pages, use the following directive:

```
User-agent: amazon-QBusiness    # Amazon Q Web Crawler
Disallow: /credential-pages/ # disallow access to specific pages
```

To allow Amazon Q Web Crawler to crawl only specific web pages, use the following directive:

```
User-agent: amazon-QBusiness   # Amazon Q Web Crawler
Allow: /pages/ # allow access to specific pages
```

To allow Amazon Q Web Crawler to crawl all website content and disallow crawling for any other robots, use the following directive:

```
User-agent: amazon-QBusiness # Amazon Q Web Crawler
Allow: / # allow access to all pages
```
Stopping Amazon Q Web Crawler from crawling your website

You can stop Amazon Q Web Crawler from indexing your website using the Disallow directive. You can also control which web pages are crawled and which aren't.

To stop Amazon Q Web Crawler from crawling the website, use the following directive:

```
User-agent: amazon-QBusiness # Amazon Q Web Crawler
Disallow: / # disallow access to any pages
```

Amazon Q Web Crawler also supports the robots noindex and nofollow directives in meta tags in HTML pages. These directives stop the web crawler from indexing a web page and stops following any links on the web page. You put the meta tags in the section of the document to specify the rules of robots rules.

For example, the below web page includes the directives robots noindex and nofollow:

```
<html>
<head>
    <meta name="robots" content="noindex, nofollow"/>
    ...
</head>
<body>...
</body>
</html>
```

If you have any questions or concerns about Amazon Q Web Crawler, you can reach out to the AWS support team.

Connecting Amazon WorkDocs to Amazon Q

Amazon WorkDocs is a secure content collaboration service for creating, editing, storing, and sharing content. Amazon Q can connect to your Amazon WorkDocs instance.

You can connect your Amazon WorkDocs instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.
Learn more

• For an overview of the Amazon Q web experience creation process, see Configuring an application.
• For an overview of connector features, see Data source connector concepts.
• For information about connector configuration best practices, see Connector configuration best practices.

Topics

• Prerequisites
• Using the console
• Using the API
• ACL crawling
• IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Amazon WorkDocs, make sure you have:

• Noted the Amazon WorkDocs directory ID (organization ID) for your Amazon WorkDocs repository.

In your AWS account, make sure you have:

• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
• Stored your Amazon WorkDocs authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Tip

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.
For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Amazon WorkDocs page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **In Source**, enter the following:
   
   - **Organization ID specific to your Amazon WorkDocs site** – Select a Amazon WorkDocs directory or create a new one. Only already created directories are available to connect.
   
   - **Amazon WorkDocs site name** – Enter your Amazon WorkDocs site name.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.
   
   **Note**
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).

4. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:
   
   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.
   
   For more information, see VPC.

5. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.
For more information, see IAM role.

6. In Sync scope – Choose what to sync from your data source.
   - Crawl document comments – Choose to crawl document comments.
   - regex patterns – Add regex patterns to include or exclude file names, file types, or file paths. You can have a total of 100 patterns.

7. In Sync mode, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.
   - Full sync – Sync all content regardless of the previous sync status.
   - New, modified, or deleted content sync – Sync only new, modified, and deleted documents.

8. In Sync run schedule, for Frequency – Choose how often Amazon Q will sync with your data source. For more details, see Sync run schedule.

9. Tags - optional – Add tags to search and filter your resources or track your AWS costs. See Tags for more details.

10. Field mappings – A list of data source document attributes to map to your index fields. Add the fields from the Data source details page after you finish adding your data source. You can choose from two types of fields:
   a. Default – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.
   b. Custom – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see Field mappings.

11. To finish connecting your data source to Amazon Q, select Add data source.

    You are taken to the Data source details, where you can view your data source configuration details.

12. In Data source details, choose Sync now to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.
**Note**

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**Amazon WorkDocs JSON schema**

The following is the Amazon WorkDocs JSON schema:

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "connectionConfiguration": {
            "type": "object",
            "properties": {
                "repositoryEndpointMetadata": {
                    "type": "object",
                    "properties": {
                        "organizationId": {
                            "type": "string",
                            "minLength": 12,
                            "maxLength": 12,
                            "pattern": "d-[0-9a-fA-F]{10}"}
                    }
                },
                "siteName": {
                    "type": "string"
                },
                "domainName": {
                    "type": "string"
                }
            }
        }
    }
}
```
{  
  "required": ["organizationId"]
}

"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": ["fieldMappings"]
}
"required": ["All"]
},
"additionalProperties": {
"type": "object",
"properties": {
"isCrawlAcl": {
  "type": "boolean"
},
"fieldForUserId": {
  "type": "string"
},
"crawlComments": {
  "type": "string"
},
"exclusionPatterns": {
  "type": "array",
  "items": {
    "type": "string"
  }
},
"inclusionPatterns": {
  "type": "array",
  "items": {
    "type": "string"
  }
}
},
"required": []
},
"enableIdentityCrawler": {
  "type": "boolean"
},
"syncMode": {
  "type": "string",
  "enum": [
    "FORCED_FULL_CRAWL",
    "CHANGE_LOG"
  ]
},
"type": {
  "type": "string",
  "pattern": "WORKDOCS"
}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>organizationId</td>
<td>The identifier of the directory corresponding to your Amazon WorkDocs site repository. You can find the organization ID in the AWS Directory Service by going to <strong>Active Directory</strong>, then <strong>Directories</strong>.</td>
</tr>
<tr>
<td>siteName</td>
<td></td>
</tr>
<tr>
<td>domainName</td>
<td></td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• All</td>
<td>A list of objects that map the attributes or field names of your Amazon WorkDocs content to Amazon Q index field names.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl ACL information.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td></td>
</tr>
<tr>
<td>crawlComments</td>
<td>Specify true to crawl pages.</td>
</tr>
<tr>
<td>• exclusionPatterns</td>
<td>A list of regular expression patterns to exclude specific content from your Amazon WorkDocs data source. Content that matches the patterns are excluded from 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.</td>
</tr>
<tr>
<td>• inclusionPatterns</td>
<td>A list of regular expression patterns to include specific content in your Amazon WorkDocs data source. Content that matches the patterns are included in the index. Content that doesn't match the patterns are excluded 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.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify WORKDOCS as your data source type.</td>
</tr>
</tbody>
</table>
## Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enableIdentityCrawler</td>
<td>Specify <strong>true</strong> to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to specific documents. See <a href="#">Identity crawler</a> for more information.</td>
</tr>
</tbody>
</table>
| syncMode                   | Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:  
  - Use `FORCED_FULL_CRAWL` to freshly re-crawl all content and replace existing content each time your data source syncs with your index.  
  - Use `CHANGE_LOG` to incrementally crawl only new and modified content each time your data source syncs with your index. |
| version                    | The version of this template that's currently supported.                    |

### ACL crawling

When you connect an Amazon WorkDocs data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Amazon WorkDocs instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

The Amazon WorkDocs group and user IDs are mapped as follows:

- `_group_ids`—Group IDs exist in Amazon WorkDocs on files where there are set access permissions. They are mapped from the names of the groups in Amazon WorkDocs.
- `_user_id`—User IDs exist in Amazon WorkDocs on files where there are set access permissions. They are mapped from the user names in Amazon WorkDocs.
IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- *(Optional)* If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToGetSecret",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:{region}:{account_id}:secret:[secret_id]"
      ]
    },
    {
      "Sid": "AllowsAmazonQToDecryptSecret",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
         "arn:aws:kms:{region}:{account_id}:alias/{kms_alias}"]
    }
  ]
}
```
"arn:aws:kms:{region}::{account_id}:key/[{key_id}]",

"Condition": {
  "StringLike": {
    "kms:ViaService": [
      "secretsmanager.*.amazonaws.com"
    ]
  }
},

{
  "Sid": "AllowsAmazonQToIngestDocuments",
  "Effect": "Allow",
  "Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
  ],
  "Resource": "arn:aws:qbusiness:{region}::{source_account}:application/{application_id}/index/{index_id}"
},

{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroups"
  ],
  "Resource": [
    "arn:aws:qbusiness:{region}::{account_id}:application/{application_id}"
  ]
},

{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "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]]"
],

{
  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:{region}::{account_id}:network-interface/*",
  "Condition": {
    "StringLike": {
      "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "AMAZON_Q"
      ]
    }
  }
},

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

{
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ]}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
        }
      }
    }
  ]
}
```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting Box to Amazon Q

Box is a cloud storage service that offers file hosting capabilities. You can connect your Box instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Box, make sure you have:

- A Box Enterprise or Box Enterprise Plus account.
• Created a Box custom app in the Box Developer Console and configured it to use **Server Authentication (with JWT)**.

• 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

• Generated and downloaded 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 authentication credentials. See [Public and private keypair](#) for more details.

• Copied your Box enterprise ID either from your Box Developer Console settings or from your Box app. For example, **801234567**.

**In your AWS account, make sure you have:**

• Created an **IAM role** for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your Box authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

**Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

**Using the console**

On the **Box** page, enter the following information:

1. **Name** – Name your data source for easy tracking.
Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Source** – Enter your **Box enterprise ID**.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   ![Note]
   
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

4. In **Authentication** – Choose to create an **AWS Secrets Manager secret** and then enter the following information for your **AWS Secrets Manager secret**.

   a. **Secret name** – A name for your secret.
   b. **Client ID** – The client ID provided by Box.
   c. **Client Secret** – The client secret provided by Box.
   d. **Public Key ID** – Your Box public key ID.
   e. **Private Key** – The private key provided by Box.
   f. **Pass Phrase** – The pass phrase you use to log into your Box account.

5. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

7. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.
8. In **Sync scope**, enter the following information:

   a. **Select additional kinds of content to index** – Choose whether to include **Web links**, **Comments**, and **Tasks**.

   b. **Additional configuration – optional** – Configure the following settings:

      • **Regex patterns** – Regular expression patterns to include or exclude certain files. You can add up to 100 patterns.

9. 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 Q for the first time, all content is synced by default.

   • **Full sync** – Sync all content regardless of the previous sync status.
   
   • **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see **Sync run schedule**.

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See **Tags** for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see **Field mappings**.

13. To finish connecting your data source to Amazon Q, select **Add data source**.
You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

**Note**

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**Box JSON schema**

The following is the Box JSON schema:

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "connectionConfiguration": {
            "type": "object",
            "properties": {
                "repositoryEndpointMetadata": {
                    "type": "object",
                    "properties": {
                        "enterpriseId": {
                            "type": "string",
                            "minLength": 9,
                            "maxLength": 9
                        }
                    }
                }
            }
        }
    }
}
```
},
  "required": [
    "enterpriseId"
  ]
},
"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",
          "dateFieldFormat"
        ]
      }
    }
  }
},
"required": [
  "indexFieldName",
  "indexFieldType",
  "dataSourceFieldName",
  "dateFieldFormat"
]
"indexFieldDataType",
  "dataSourceFieldName"
]
]}
}
"required": [
  "fieldMappings"
],
"task": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "indexFieldName": {
              "type": "string"
            },
            "indexFieldDataType": {
              "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",
          "indexFieldDataType",
          "dataSourceFieldName"
```


]}
"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"
]

"additionalProperties": {
  "type": "object",
  "properties": {
    "isCrawlAcl": {
      "type": "boolean"
    },
    "fieldForUserId": {
      "type": "string"
    },
    "crawlComments": {
      "type": "boolean"
    },
    "crawlTasks": {
      "type": "boolean"
    },
    "crawlWebLinks": {
      "type": "boolean"
    },
    "inclusionPatterns": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "exclusionPatterns": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  },
  "required": []
},
"type": {
  "type": "string",
  "pattern": "BOX"}
"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",
  "syncMode",
  "additionalProperties",
  "secretArn",
  "version",
  "enableIdentityCrawler"
]}

The following table provides information about important JSON keys to configure.
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>enterpriseId</td>
<td>The Box enterprise id.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• file</td>
<td>A list of objects that map the attributes or field names of your Box files, tasks, comments, and webLinks to Amazon Q index field names.</td>
</tr>
<tr>
<td>• task</td>
<td></td>
</tr>
<tr>
<td>• comment</td>
<td></td>
</tr>
<tr>
<td>• webLink</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>crawlComments</td>
<td>Specify true to crawl assets.</td>
</tr>
<tr>
<td>crawlTasks</td>
<td>Specify true to crawl pages.</td>
</tr>
<tr>
<td>crawlWebLinks</td>
<td>Specify true to crawl pages.</td>
</tr>
<tr>
<td>• InclusionPatterns</td>
<td>A list of regular expression patterns to include or exclude specific content from your Box 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</td>
</tr>
<tr>
<td>• ExclusionPatterns</td>
<td></td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>and exclusion pattern</td>
<td>and exclusion pattern takes precedence, and the content isn't included in the index.</td>
</tr>
<tr>
<td><strong>type</strong></td>
<td>The type of data source. Specify BOX as your data source type.</td>
</tr>
<tr>
<td><strong>enableIdentityCrawler</strong></td>
<td>Specify true to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to specific documents.</td>
</tr>
<tr>
<td><strong>syncMode</strong></td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>• Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</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 Box. The secret must contain a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>`{</td>
</tr>
</tbody>
</table>
|               |   "clientID": "client-id",
|               |   "clientSecret": "client-secret",
|               |   "publicKeyId": "public-key-id",
|               |   "privateKey": "private-key",
|               |   "passphrase": "pass-phrase"
|               | } |
| version       | The version of this template that's currently supported. |

**ACL crawling**

When you connect an Box data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Box instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

- `_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.
- `_userId`—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.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)
IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToGetSecret",
            "Effect": "Allow",
            "Action": ["secretsmanager:GetSecretValue"],
            "Resource": [
                "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:{{secret_id}}"
            ]
        },
        {
            "Sid": "AllowsAmazonQToDecryptSecret",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:{{region}}:{{account_id}}:key:{{key_id}}"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": ["secretsmanager.*.amazonaws.com"]
                }
            }
        }
    ]
}
```
{}},
},
{
  "Sid": "AllowsAmazonQToIngestDocuments",
  "Effect": "Allow",
  "Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
  ],
  "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},
{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroups"
  ],
  "Resource": [
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**"
  ]}
},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "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]]"
  ]
}]
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
    "StringLike": {
        "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
        "aws:TagKeys": [
            "AMAZON_Q"
        ]
    }
},
],
{
    "Sid": "AllowsAmazonQToCreateTags",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "CreateNetworkInterface"
        }
    }
},
{
    "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterfacePermission"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
    "Condition": {
        "StringLike": {
            "aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
        }
    }
},
]
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQServicePrincipal",
            "Effect": "Allow",
            "Principal": {
                "Service": "qbusiness.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "{{source_account}}"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
                }
            }
        }
    ]
}
```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

**Connecting Confluence (Cloud) to Amazon Q**

Atlassian Confluence is a collaborative work-management tool designed for sharing, storing, and working on project planning, software development, and product management. You can connect Confluence (Cloud) instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

**Learn more**

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

**Topics**

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

**Prerequisites**

Before you begin, make sure that you have completed the following prerequisites.

**In Confluence Cloud, make sure you have:**

- Copied your Confluence instance URL. For example: https://example.confluence.com. You need your Confluence instance URL to connect to Amazon Q.
- Configured basic authentication credentials containing a username (email ID used to log into Confluence) and password (Confluence API token) to allow Amazon Q to connect to your Confluence instance. For information about how to create a Confluence API token, see Manage API tokens for your Atlassian account on the Atlassian website.
• **Optional:** Configured OAuth 2.0 credentials containing a Confluence app key, Confluence app secret, Confluence access token, and Confluence refresh token to allow Amazon Q 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 about access tokens, see [Manage OAuth access tokens](#) on the Atlassian website.

**In your AWS account, make sure you have:**

• Created an [IAM role](#) for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your Confluence (Cloud) authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

**Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

**Using the console**

On the **Confluence** page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In **Source**, enter the following information:
   
   a. In **Source**, for **Hosting Method**, choose **Confluence Cloud**.
   
   b. **Confluence URL** – Enter the Confluence host URLs. The format for the host URL that you enter is `https://example.confluence.com`.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.
4. For Authentication – Choose between Basic authentication and Oauth 2.0 authentication, based on your use case.

5. AWS Secrets Manager secret – Choose an existing secret or create a Secrets Manager secret to store your Confluence authentication credentials. If you choose to create a secret, an AWS Secrets Manager secret window opens. Enter the following information in the window:
   a. Secret name – A name for your secret.
   b. If using Basic Authentication – Enter the Secret name, User name, and Password (Confluence API token) that you generated and downloaded from your Confluence account.
      If using OAuth2.0 Authentication – Enter the Secret name, App key, App secret, Access token, and Refresh token that you created in your Confluence account.
   c. Choose Save and add secret.

6. Configure VPC and security group – optional – Choose whether you want to use a VPC. If you do, enter the following information:
   a. Subnets – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   b. VPC security groups – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.
      For more information, see VPC.

7. Identity crawler – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see Identity crawler.

8. IAM role – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.
      For more information, see IAM role.
9. **In Sync scope**, choose from the following options:

   a. **In Sync scope**, for **sync contents**, choose to sync from the following entity types: **Pages**, Page comments, Page attachments, Blogs, Blog comments, Blog attachments, Personal spaces Archived spaces, and Archived pages.

      - **Note**
        Page comments and Page attachments can only be selected if you choose to sync Pages. Blog comments and Blog attachments can only be selected 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 – optional**, for **Space and 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 in Additional configuration, 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.
10. 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 Q for the first time, all content is synced by default.

   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

11. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see **Sync run schedule**.

12. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See **Tags** for more details.

13. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see **Field mappings**.

14. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

15. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.
Note

You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

Using the API

You use the CreateDataSource action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

Confluence JSON schema

The following is the Confluence JSON schema:

```json
{
   "$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": ["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"
                                }
                            }
                        }
                    ]
                }
            }
        }
    }
}
,"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"
],
"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"
            }
          }
        }
      ]
    }
  }
}
"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'"
            }
          }
        }
      ]
    }
  }
}
```
{
    "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"
            ]
        }
    },
    "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",
                        "dateFieldFormat"
                    ]
                }
            }
        }
    }
}
"indexFieldType",
"dataSourceFieldName"
]

"additionalProperties": {
  "type": "object",
  "properties": {
    "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"
},
"inclusionFileTypePatterns": {
  "type": "array",
  "items": {"type": "string"}
"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"
  ]
},
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>hostUrl</td>
<td>The URL for your Confluence instance. For example, <a href="https://example.confluence.com">https://example.confluence.com</a> .</td>
</tr>
<tr>
<td>type</td>
<td>The hosting method for your Confluence instance, whether SAAS or ON_PREM.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>authType</td>
<td>The authentication method for your Confluence instance, whether Basic, OAuth2, or Personal-token.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• space</td>
<td>A list of objects that map the attributes or field names of your Confluence spaces, pages, blogs, comments, and attachments to Amazon Q index field names.</td>
</tr>
<tr>
<td>• page</td>
<td></td>
</tr>
<tr>
<td>• blog</td>
<td></td>
</tr>
<tr>
<td>• comment</td>
<td></td>
</tr>
<tr>
<td>• attachment</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>inclusionSpaceKeyFilter</td>
<td>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.</td>
</tr>
<tr>
<td>exclusionSpaceKeyFilter</td>
<td></td>
</tr>
<tr>
<td>pageTitleRegEX</td>
<td></td>
</tr>
<tr>
<td>blogTitleRegEX</td>
<td></td>
</tr>
<tr>
<td>commentTitleRegEX</td>
<td></td>
</tr>
<tr>
<td>attachmentTitleRegEX</td>
<td></td>
</tr>
<tr>
<td>inclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionUrlPatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionUrlPatterns</td>
<td></td>
</tr>
<tr>
<td>proxyHost</td>
<td></td>
</tr>
<tr>
<td>proxyPort</td>
<td></td>
</tr>
<tr>
<td>fieldForUserId</td>
<td></td>
</tr>
<tr>
<td>isCrawlPersonalSpace</td>
<td>true to index files in your Confluence personal spaces, pages, blogs, page comments, page attachments, blog comments, and blog attachments.</td>
</tr>
<tr>
<td>isCrawlArchivedSpace</td>
<td></td>
</tr>
<tr>
<td>isCrawlArchivedPage</td>
<td></td>
</tr>
<tr>
<td>isCrawlPage</td>
<td></td>
</tr>
<tr>
<td>isCrawlBlog</td>
<td></td>
</tr>
<tr>
<td>isCrawlBlogComment</td>
<td></td>
</tr>
<tr>
<td>isCrawlPageComment</td>
<td></td>
</tr>
<tr>
<td>isCrawlPageAttachment</td>
<td></td>
</tr>
<tr>
<td>isCrawlBlogComment</td>
<td></td>
</tr>
<tr>
<td>isCrawlBlogAttachment</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify CONFLUENCE EV2 as your data source type.</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>true to activate identity crawler. Identity crawler is activated by default. See Identity crawler for more information.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>secretARN</td>
<td>The Amazon Resource Name (ARN) of a Secrets Manager secret that contains the key-value pairs required to connect to your Confluence instance. If you use basic authentication, the secret must contain a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;username&quot;: &quot; <em>Confluence account user name</em> &quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;password&quot;: &quot; <em>Confluence API token</em> &quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>If you use OAuth 2.0 authentication, the secret must contain a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;confluenceAppKey&quot;: &quot; <em>app key for your Confluence account</em> &quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;confluenceAppSecret&quot;: &quot; <em>app secret from your Confluence token</em> &quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;confluenceAccessToken&quot;: &quot; <em>access token created in Confluence</em> &quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;confluenceRefreshToken&quot;: &quot; <em>refresh token created in Confluence</em> &quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>(For Confluence Server only) If you use basic authentication, the secret is stored in a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;hostUrl&quot;: &quot; <em>Confluence Server host URL</em> &quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Description</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;username&quot;: &quot;Confluence Server user name&quot;</td>
<td>&quot;username&quot;: &quot;Confluence Server user name&quot;</td>
</tr>
<tr>
<td>&quot;password&quot;: &quot;Confluence Server password&quot;</td>
<td></td>
</tr>
</tbody>
</table>

(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": "Confluence token"
}
```

| version | The version of this template that's currently supported. |

### ACL crawling

When you connect an Confluence data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Confluence instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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 group and user IDs are mapped as follows:

- _group_ids – Group names are present on spaces, pages, and blogs where there are restrictions. They're 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’re mapped depending on the type of Confluence instance that you are using.

• For Confluence Cloud – The `_user_id` is the account 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](https://developer.atlassian.com/cloud/confluence/visibility/) in Atlassian Developer Documentation.

For more information, see:

• [Authorization](#)

• [Identity crawler](#)

• [Understanding User Store](#)

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

• Permission to access the `BatchPutDocument` and `BatchDeleteDocument` operations to ingest documents.

• Permission to access the [User Store](https://developer.atlassian.com/cloud/confluence/user-store/) API operations to ingest user and group access control information from documents.

• Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.

• *(Optional)* If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToGetSecret",
            "Effect": "Allow",
```
"Action": [
   "secretsmanager:GetSecretValue"
],
"Resource": [
   "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
]
},
{
   "Sid": "AllowsAmazonQToDecryptSecret",
   "Effect": "Allow",
   "Action": [
      "kms:Decrypt"
   ],
   "Resource": [
      "arn:aws:kms:{{region}}:{{account_id}}:key/[[key_id]]"
   ],
   "Condition": {
      "StringLike": {
         "kms:ViaService": [
            "secretsmanager.*.amazonaws.com"
         ]
      }
   }
},
{
   "Sid": "AllowsAmazonQToIngestDocuments",
   "Effect": "Allow",
   "Action": [
      "qbusiness:BatchPutDocument",
      "qbusiness:BatchDeleteDocument"
   ],
   "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},
{
   "Sid": "AllowsAmazonQToIngestPrincipalMapping",
   "Effect": "Allow",
   "Action": [
      "qbusiness:PutGroup",
      "qbusiness:CreateUser",
      "qbusiness:DeleteGroup",
      "qbusiness:UpdateUser",
      "qbusiness:ListGroups"
   ],
"Resource": [  
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",  
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/  
  index/{{index_id}}",  
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/  
  index/{{index_id}}/data-source/*"  
],

{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "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]]"
  ]
},

{
  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
  "Effect": "Allow",
  "Action": [  
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [  
        "AMAZON_Q"
      ]
    }
  }
},

{
  "Sid": "AllowsAmazonQToCreateTags",
  "Effect": "Allow",
  "Action": [  
    "ec2:CreateTags"
  ]
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
      "Effect": "Allow",
      "Action": ["ec2:CreateNetworkInterfacePermission"],
      "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
        }
      }
    },
    {
      "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
      "Effect": "Allow",
      "Action": ["ec2:DescribeNetworkInterfaces",
                 "ec2:DescribeAvailabilityZones",
                 "ec2:DescribeNetworkInterfaceAttribute",
                 "ec2:DescribeVpcs",
                 "ec2:DescribeRegions",
                 "ec2:DescribeNetworkInterfacePermissions",
                 "ec2:DescribeSubnets"
               ],
      "Resource": "*"
    }
  ]
}
For more information on Amazon Q data source connector IAM roles, see [IAM roles for Amazon Q data source connectors](#).

**Connecting Confluence (Server) to Amazon Q**

Atlassian Confluence is a collaborative work-management tool designed for sharing, storing, and working on project planning, software development, and product management. You can connect Confluence (Server) instance to Amazon Q—using either the AWS Management Console or the `CreateDataSource` API—and create an Amazon Q web experience.

**Learn more**

- For an overview of the Amazon Q web experience creation process, see [Configuring an application](#).
- For an overview of connector features, see [Data source connector concepts](#).
- For information about connector configuration best practices, see [Connector configuration best practices](#).

**Topics**

- **Prerequisites**
• Using the console
• Using the API
• ACL crawling
• IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Confluence Server, make sure you have:

• Copied your Confluence instance URL. For example: https://example.confluence.com. You need your Confluence instance URL to connect to Amazon Q.

• Configured basic authentication credentials containing a username (email ID used to log into Confluence) and password (Confluence Server password) to allow Amazon Q to connect to your Confluence Server instance. For information about how to create a Confluence API token, see Manage API tokens for your Atlassian account on the Atlassian website.

• Optional: Configured OAuth 2.0 credentials containing a Confluence app key, Confluence app secret, Confluence access token, and Confluence refresh token to allow Amazon Q 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.

• Optional: Configured a Personal Access Token (PAT) containing a Confluence token to allow Amazon Q to connect to your Confluence Server instance. For information about how to create a PAT token, see Using Personal Access Tokens on the Atlassian website.

In your AWS account, make sure you have:

• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your Confluence (Server) authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Confluence page, enter the following information:

1. Name – Name your data source for easy tracking.

   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In Source, enter the following information:
   a. In Source, for Hosting Method – Choose Confluence Server.
   b. Confluence URL – Enter the Confluence host URLs. The format for the host URL that you enter is https://example.confluence.com.
   c. SSL certificate location – Enter the file path to an SSL certificate stored in an Amazon S3 bucket.

3. Web proxy – optional, enter the following information:
   a. Host name – Host name for your Confluence account.
   b. Port number – Port used by the host URL transport protocol.

4. Authorization – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.

   Note
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).
5. For **Authentication** – Choose between **Basic authentication**, **Oauth 2.0 authentication**, and **Personal Access Token authentication** based on your use case.

6. **AWS Secrets Manager secret** – Choose an existing secret or create a Secrets Manager secret to store your Confluence authentication credentials. If you choose to create a secret, an AWS Secrets Manager secret window opens. Enter the following information in the window:
   
   a. **Secret name** – A name for your secret.
   
   b. If using **Basic Authentication** – Enter the **Secret name Username**, and **Password** (Confluence Server password) that you generated and downloaded from your Confluence account.

   If using **OAuth2.0 Authentication** – Enter the **Secret name**, **App key**, **App secret**, **Access token**, and **Refresh token** you created in your Confluence account.

   If using **Personal Access Token authentication** – Enter the **Secret name** and **Confluence token** that you created in your Confluence account.

   c. Choose **Save and add secret**.

7. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

8. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

9. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

10. In **Sync scope**, choose from the following options:

**Note**

**Page comments** and **Page attachments** can only be selected if you choose to sync **Pages**. **Blog comments** and **Blog attachments** can only be selected 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 – optional**, for **Space and 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 in **Additional configuration**, 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.
11. 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 Q for the first time, all content is synced by default.

- **Full sync** – Sync all content regardless of the previous sync status.
- **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

12. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

13. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

14. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

15. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

16. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.
Note
You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

Using the API

You use the CreateDataSource action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

Confluence JSON schema

The following is the Confluence JSON schema:

```json
{
  "$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", "OAuth"]
            }
          }
        }
      }
    }
  }
}
```
"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"
    ]
},
"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"
                        }
                    }
                }
            ]
        }
    }
}
"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'"


```json
{
  "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"
  ]
}
}
"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",
            "dateFieldFormat"
          ]
        }
      ]
    }
  }
}
"indexFieldType",
"dataSourceFieldName"
]
}
]
},
"required": [
  "fieldMappings"
]
}
},
"additionalProperties": {
  "type": "object",
  "properties": {
    "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"
},
"isCrawlBlog": {
    "type": "boolean"
},
"isCrawlPage": {
    "type": "boolean"
},
"isCrawlBlogComment": {
    "type": "boolean"
},
"isCrawlPageComment": {
    "type": "boolean"
},
"isCrawlPageAttachment": {
    "type": "boolean"
},
"isCrawlBlogAttachment": {
    "type": "boolean"
},
"inclusionFileTypePatterns": {
    "type": "array",
    "items": {
        "type": "string"
    }
}
"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"
]
}

The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>hostUrl</td>
<td>The URL for your Confluence instance. For example, <a href="https://example.confluence.com">https://example.confluence.com</a>.</td>
</tr>
<tr>
<td>type</td>
<td>The hosting method for your Confluence instance, whether SAAS or ON_PREM.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>authType</td>
<td>The authentication method for your Confluence instance, whether Basic, OAuth2, or Personal-token.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td></td>
<td>• space • page • blog • comment • attachment</td>
</tr>
<tr>
<td></td>
<td>A list of objects that map the attributes or field names of your Confluence spaces, pages, blogs, comments, and attachments to Amazon Q index field names.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• inclusionSpaceKeyFilter</td>
<td>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.</td>
</tr>
<tr>
<td>• exclusionSpaceKeyFilter</td>
<td></td>
</tr>
<tr>
<td>• pageTitleRegEX</td>
<td></td>
</tr>
<tr>
<td>• blogTitleRegEX</td>
<td></td>
</tr>
<tr>
<td>• commentTitleRegEX</td>
<td></td>
</tr>
<tr>
<td>• attachmentTitleRegEX</td>
<td></td>
</tr>
<tr>
<td>• inclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• exclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionUrlPatterns</td>
<td></td>
</tr>
<tr>
<td>• exclusionUrlPatterns</td>
<td></td>
</tr>
<tr>
<td>• proxyHost</td>
<td></td>
</tr>
<tr>
<td>• proxyPort</td>
<td></td>
</tr>
<tr>
<td>• fieldForUserId</td>
<td></td>
</tr>
<tr>
<td>• isCrawlPersonalSpace</td>
<td>true to index files in your Confluence personal spaces, pages, blogs, page comments, page attachments, blog comments, and blog attachments.</td>
</tr>
<tr>
<td>• isCrawl ArchivedSpace</td>
<td></td>
</tr>
<tr>
<td>• isCrawl ArchivedPage</td>
<td></td>
</tr>
<tr>
<td>• isCrawlPage</td>
<td></td>
</tr>
<tr>
<td>• isCrawlBlog</td>
<td></td>
</tr>
<tr>
<td>• isCrawlPageComment</td>
<td></td>
</tr>
<tr>
<td>• isCrawlPageAttachment</td>
<td></td>
</tr>
<tr>
<td>• isCrawlBlogComment</td>
<td></td>
</tr>
<tr>
<td>• isCrawlBlogAttachment</td>
<td></td>
</tr>
<tr>
<td>• isCrawlBlogComment</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify CONFLUENCE EV2 as your data source type.</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>true to activate identity crawler. Identity crawler is activated by default. See Identity crawler for more information.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use <code>FORCED_FULL_CRAWL</code> to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use <code>FULL_CRAWL</code> to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>secretARN</td>
<td>The Amazon Resource Name (ARN) of a Secrets Manager secret that contains the key-value pairs required to connect to your Confluence instance. If you use basic authentication, the secret must contain a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>{ &quot;username&quot;: &quot;Confluence account username&quot;, &quot;password&quot;: &quot;Confluence API token&quot; }</td>
</tr>
<tr>
<td></td>
<td>If you use OAuth 2.0 authentication, the secret must contain a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>{ &quot;confluenceAppKey&quot;: &quot;app key for your Confluence account &quot;, &quot;confluenceAppSecret&quot;: &quot;app secret from your Confluence token &quot;, &quot;confluenceAccessToken&quot;: &quot;access token created in Confluence &quot;, &quot;confluenceRefreshToken&quot;: &quot;refresh token created in Confluence &quot; }</td>
</tr>
<tr>
<td></td>
<td>(For Confluence Server only) If you use basic authentication, the secret is stored in a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>{ &quot;hostUrl&quot;: &quot;Confluence Server host URL&quot; }</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>&quot;username&quot;: &quot;Confluence Server user name&quot;, &quot;password&quot;: &quot;Confluence Server password&quot;</td>
<td>(For Confluence Server only) If you use Personal Access Token authentication, the secret is stored in a JSON structure with the following keys:</td>
</tr>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>

**ACL crawling**

When you connect an Confluence data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Confluence instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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 group and user IDs are mapped as follows:

- **_group_ids** – Group names are present on spaces, pages, and blogs where there are restrictions. They're 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’re mapped depending on the type of Confluence instance that you are using.

• For Confluence Cloud – The _user_id is the account 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.

For more information, see:

• Authorization
• Identity crawler
• Understanding User Store

IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

• Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
• Permission to access the User Store API operations to ingest user and group access control information from documents.
• Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
• Permission to access the SSL certificate stored in your Amazon S3 bucket.
• (Optional) If you’re using Amazon VPC, permission to access your Amazon VPC.

```json
{
   "Version": "2012-10-17",
   "Statement": [ {
      "Sid": "AllowsAmazonQToGetS3Objects",
      "Action": [ ]
   }]
}```
"s3:GetObject",
  "Resource": [
    "arn:aws:s3:::{{input_bucket_name}}/*"
  ],
  "Effect": "Allow",
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "{{account_id}}"
    }
  }
],
{"Sid": "AllowsAmazonQToGetSecret",
 "Effect": "Allow",
 "Action": [
   "secretsmanager:GetSecretValue"
 ],
 "Resource": [
   "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:{{[secret_id]}}"
 ]
},
{"Sid": "AllowsAmazonQToDecryptSecret",
 "Effect": "Allow",
 "Action": [
   "kms:Decrypt"
 ],
 "Resource": [
   "arn:aws:kms:{{region}}:{{account_id}}:key/{{[key_id]}}"
 ],
 "Condition": {
   "StringLike": {
     "kms:ViaService": [
       "secretsmanager.*.amazonaws.com"
     ]
   }
 }
},
{"Sid": "AllowsAmazonQToIngestDocuments",
 "Effect": "Allow",
 "Action": [
   "qbusiness:BatchPutDocument",
   "s3:PutObject"
 ],
 "Resource": [
   "arn:aws:s3:::{{input_bucket_name}}/{{{{project_id}}}}/index.json"
 ],
 "Condition": {
   "StringLike": {
     "s3:Prefix": [
       "{{[bucket_prefix]}}/index.json"
     ]
   }
 }
}
"qbusiness:BatchDeleteDocument"
],
"Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},
{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroups"
  ],
  "Resource": [
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**"
  ],
},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "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]]"
  ],
},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
  ]}
"Resource": "arn:aws:ec2:{region}::{account_id}:network-interface/**",
"Condition": {
  "StringLike": {
    "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_**",
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "AMAZON_Q"
      ]
    }
  }
},
{
  "Sid": "AllowsAmazonQToCreateTags",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": "arn:aws:ec2:{region}::{account_id}:network-interface/**",
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
    }
  }
},
{
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ],
  "Resource": "arn:aws:ec2:{region}::{account_id}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_**"
    }
  }
},
{
  "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
  "Effect": "Allow",
  "Action": [
"ec2:DescribeNetworkInterfaces",
"ec2:DescribeAvailabilityZones",
"ec2:DescribeNetworkInterfaceAttribute",
"ec2:DescribeVpcs",
"ec2:DescribeRegions",
"ec2:DescribeNetworkInterfacePermissions",
"ec2:DescribeSubnets"
]

"Resource": "*"

}

To allow Amazon Q to assume a role, you must also use the following trust policy:

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
        }
      }
    }
  ]
}

For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.
Connecting Dropbox to Amazon Q

Dropbox is a file hosting service that offers cloud storage, document organization, and document templating services. You can connect Dropbox instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

• For an overview of the Amazon Q web experience creation process, see Configuring an application.

• For an overview of connector features, see Data source connector concepts.

• For information about connector configuration best practices, see Connector configuration best practices.

Topics

• Prerequisites

• Using the console

• Using the API

• ACL crawling

• IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Dropbox, make sure you have:

• Created a Dropbox Advanced account and set up an admin user.

• Created a Dropbox app with a unique App name, activated Scoped Access. For more information, see Dropbox documentation on creating an app on the Dropbox website.

• 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.

• Generated and copied a temporary Oauth 2.0 access token for your Dropbox app. This token is temporary and expires after 4 hours. For more information, see Dropbox documentation on OAuth authentication on the Dropbox website.

  **Recommended:** Configured a Dropbox permanent refresh token that never expires to allow Amazon Q to continue to sync your data source without any disruptions. For more information, see Dropbox documentation on refresh tokens on the Dropbox website.

**In your AWS account, make sure you have:**

• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your Dropbox authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

**Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

**Using the console**

On the Dropbox page, enter the following information:

1. **Name** – Name your data source for easy tracking.

   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.
2. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   ![Note](image)
   
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

3. In **Authentication** – Choose between **Permanent Token (recommended)** and **Access Token (temporary use)** based on your use case.

4. In **Authentication credentials**, for **AWS Secrets Manager secret** – Choose an existing secret or create a Secrets Manager secret to store your Dropbox authentication credentials. If you choose to create a 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.
     
     ii. For **App key**, **App secret**, and token information (permanent or temporary) – Enter the authentication credential values that you generated from your Dropbox account.
     
     iii. Choose **Save**.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.
For more information, see IAM role.

8. In **Sync scope**, enter the following information.
   a. For **Select entities or content types** – Choose entities or content types you want to crawl.
   b. **Change log mode** – Choose to update your index instead of syncing all files.
   c. In **Additional configuration – optional**, for **Regex patterns** – Add regular expression patterns to include or exclude certain files.

9. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see Sync run schedule.

10. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See Tags for more details.

11. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:
   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.
   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see Field mappings.

12. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

13. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

    **Note**
    You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to
view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

Using the API

You use the `CreateDataSource` action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

Dropbox JSON schema

The following is the Dropbox JSON schema:

```json
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "type": "object",
   "required": [
                         "repositoryEndpointMetadata"
   ],
   "properties": {
       "connectionConfiguration": {
           "type": "object",
           "properties": {
               "repositoryEndpointMetadata": {
                   "type": "object",
                   "properties": {

                       
                   }
               }
           }
       }
   },
   "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": [
          {
            "$ref": "http://example.com"
          }
        ]
      }
    }
  }
}
"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"
]
"fieldMappings": {
    "type": "object",
    "properties": {
        "properties": {
            "type": "array",
            "items": {
                "anyOf": [
                    {}]
                ]
            }
        }
    }
},
"papert": {
    "type": "object",
    "properties": {
        "properties": {
            "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"
]
}
}
"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"
]
},
"secretArn": {
  "type": "string"
},
"enableIdentityCrawler": {
  "type": "boolean"
},
"additionalProperties": {
  "type": "object",
  "properties": {
    "isCrawlAcl": {
      "type": "boolean"
    },
    "fieldForUserId": {
      "type": "string"
    },
    "inclusionPatterns": {
      "type": "array"
    },
    "exclusionPatterns": {
      "type": "array"
    },
    "crawlFile": {
      "type": "boolean"
    },
    "crawlPaper": {
      "type": "boolean"
    },
    "crawlPapert": {
      "type": "boolean"
    },
    "crawlShortcut": {
      "type": "boolean"
    }
  }
},
"type": {
  "type": "string",
  "pattern": "DROPBOX"
},
"useChangeLog": {
  "type": "string",
  "enum": [
    "true",
    "false"
  ]
},
"tokenType": {
  "type": "string",
  "enum": [
    "PERMANENT",
    "DROPBOX"
  ]
}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source. This data source doesn't specify an endpoint in repositoryEndpointMetadata. Rather, the connection information is included in an AWS Secrets Manager secret that you provide the secretArn.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>file, paper, papert,</td>
<td>A list of objects that map the attributes or field names of your Dropbox files, Dropbox Paper, and shortcuts to Amazon Q index field names.</td>
</tr>
<tr>
<td>shortcut</td>
<td></td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>Specify <code>true</code> to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to specific documents.</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 Dropbox. The secret must contain a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;appKey&quot;: &quot;Dropbox app key&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;appSecret&quot;: &quot;Dropbox app secret&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;accesstoken&quot;: &quot;temporary access token or refresh token&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify <code>true</code> to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for <code>UserId</code> for ACL crawling.</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>inclusionFileTypePatterns</td>
<td>A list of regular expression patterns to <em>include</em> specific file 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 precedence and the file isn't included in the index.</td>
</tr>
<tr>
<td>exclusionFileTypePatterns</td>
<td>A list of regular expression patterns to <em>exclude</em> specific file 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.</td>
</tr>
<tr>
<td>exclusionFileNamePatterns</td>
<td>A list of regular expression patterns to <em>exclude</em> specific file names 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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>exclusionFileNamePatterns</td>
<td>A list of regular expression patterns to exclude specific file names 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.</td>
</tr>
<tr>
<td>• crawl1File</td>
<td>true to index files in your Dropbox, Dropbox Paper documents, Dropbox Paper templates, and webpage shortcuts stored in your Dropbox.</td>
</tr>
<tr>
<td>• crawl1Paper</td>
<td></td>
</tr>
<tr>
<td>• crawl1Papert</td>
<td></td>
</tr>
<tr>
<td>• crawl1Shortcut</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify DROPBOX as your data source type.</td>
</tr>
<tr>
<td>useChangeLog</td>
<td>true to use the Dropbox change log to determine which documents require adding, updating, or deleting in the index. Depending on the change log’s size, it may take longer for Amazon Q to use the change log than to scan all of your documents in your Dropbox.</td>
</tr>
<tr>
<td>tokenType</td>
<td>Specify your access token type: permanent or temporary access token. We recommend that you create a refresh access token that never expires in Dropbox rather than 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.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>

## ACL crawling

When you connect an Dropbox data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Dropbox instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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're 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're mapped from the user emails as the IDs in Dropbox.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)

## IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the `BatchPutDocument` and `BatchDeleteDocument` operations to ingest documents.
- Permission to access the [User Store](#) API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToGetSecret",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[{{secret_id}}]"
      ]
    },
    {
      "Sid": "AllowsAmazonQToDecryptSecret",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:{{region}}:{{account_id}}:key:[{{key_id}}]"
      ],
      "Condition": {
        "StringLike": {
          "kms:ViaService": [
            "secretsmanager.*.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AllowsAmazonQToIngestDocuments",
      "Effect": "Allow",
      "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
      ],
      "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
    },
    {
      "Sid": "AllowsAmazonQToIngestPrincipalMapping",
      "Effect": "Allow",
      "Action": [
        "qbusiness:IngestPrincipalMapping"
      ],
      "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
    }
  ]
}
"Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroup"
],
"Resource": [
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**"
],
{
    "Sid": "AllowsAmazonQToCreateAndDeleteNI",
    "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]]"
    ]
},
{
    "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
    "Condition": {
        "StringLike": {
            "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
        },
        "ForAllValues:StringEquals": {
            "aws:TagKeys": [
                "AMAZON_Q"
            ]
        }
    }
}


```json
}
},
{
"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
  "ec2:CreateTags"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
  "StringEquals": {
    "ec2:CreateAction": "CreateNetworkInterface"
  }
}
},
{
"Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
"Effect": "Allow",
"Action": [
  "ec2:CreateNetworkInterfacePermission"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
  "StringLike": {
    "aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}__*"
  }
}
},
{
"Sid": "AllowsAmazonQToDescribeResourcesForVPC",
"Effect": "Allow",
"Action": [
  "ec2:DescribeNetworkInterfaces",
  "ec2:DescribeAvailabilityZones",
  "ec2:DescribeNetworkInterfaceAttribute",
  "ec2:DescribeVpcs",
  "ec2:DescribeRegions",
  "ec2:DescribeNetworkInterfacePermissions",
  "ec2:DescribeSubnets"
],
"Resource": "**"
}
]
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowsAmazonQServicePrincipal",
         "Effect": "Allow",
         "Principal": {
            "Service": "qbusiness.amazonaws.com"
         },
         "Action": "sts:AssumeRole",
         "Condition": {
            "StringEquals": {
               "aws:SourceAccount": "{{source_account}}"
            },
            "ArnEquals": {
               "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
            }
         }
      }
   ]
}
```

For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

### Connecting Drupal to Amazon Q

Drupal is an open-source content management system (CMS) that you can use to create websites and web applications. You can connect Drupal instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

**Learn more**

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
• For information about connector configuration best practices, see Connector configuration best practices.

**Topics**

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

**Prerequisites**

Before you begin, make sure that you have completed the following prerequisites.

**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 Q.
- 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 IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
• Stored your Drupal authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Drupal page, enter the following information:

1. Name – Name your data source for easy tracking.
Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In **Source**, for **Host URL** – Enter the host URL of your Drupal site. For example, `https://<hostname>/<drupalsitename>`.

3. **SSL certificate location** – Enter the path to the SSL certificate stored in an Amazon S3 bucket. You use this to connect to Drupal with a secure SSL connection.

4. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See **Authorization** for more details.

   \(\text{Note}\) Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see **Integrating with an Identity Provider (IdP)**.

5. **Authentication** – Choose between **Basic authentication** and **OAuth 2.0 authentication** and then enter the following information for your **AWS Secrets Manager secret**.
   a. **Basic authentication** – Enter the **User name**, (Drupal site user name), and **Password** (Drupal site password).
   b. **OAuth 2.0 authentication** – Enter the **User name**, (Drupal site user name), and **Password** (Drupal site password).

6. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:
   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see **VPC**.

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see **IAM role**.
8. In **Sync scope**, enter the following information:

   a. 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 and crawl **Custom content types** and **Custom Blocks**.

   b. (Optional) **Additional configuration** – Configure the following settings:
      
      - **Regex pattern**—Add regular expression patterns to include or exclude specific entity titles and file names. You can add up to 100 patterns.

9. 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 Q for the first time, all content is synced by default.

   - **Full sync**—Sync all content regardless of the previous sync status.
   - **New, modified, or deleted content sync**—Sync only new, modified, and deleted documents.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.
You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

**Note**

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

### Using the API

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**Drupal JSON schema**

The following is the Drupal JSON schema:

```json
{
    "$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": [
  "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": {
    "isCrawlAcl": {
      "type": "boolean"
    },
    "fieldForUserId": {
      "type": "string"
    },
    "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": """"n"
        }
      ]
    }
  }
},
"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"
    }
}
"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"
},
"contentType",
"fieldDefinition",
"isCrawlComments",
"isCrawlFiles"
],
"required": []},
"type": {
  "type": "string",
  "pattern": "DRUPAL"
},
"authType": {
  "type": "string",
  "enum": [
    "BASIC-AUTH",
    "OAUTH2"
  ]},
"syncMode": {
  "type": "string",
  "enum": [
    "FORCED_FULL_CRAWL",
    "FORCEFULLCRAWL"
  ]}
The following provides information on important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>hostUrl</td>
<td>The host URL of your Drupal website. For example, https://&lt;hostname&gt;/&lt;drupalsitename&gt;.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source.</td>
</tr>
<tr>
<td>• content</td>
<td>A list of objects that map the attributes or field names of your Drupal files. The Drupal data source field names must exist in your Drupal custom metadata.</td>
</tr>
<tr>
<td>• comment</td>
<td></td>
</tr>
<tr>
<td>• attachment</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>• inclusionFileNamePatterns</td>
<td>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.</td>
</tr>
<tr>
<td>• articleTitleInclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• pageTitleInclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• customContentTitleInclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• basicBlockTitleInclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• customBlockTitleInclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• exclusionFileNamePatterns</td>
<td>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.</td>
</tr>
<tr>
<td>• articleTitleExclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• pageTitleExclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• customContentTitleExclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• basicBlockTitleExclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>• customBlockTitleExclusionPatterns</td>
<td></td>
</tr>
<tr>
<td>contentDefinitions</td>
<td>Specify the content types to crawl and whether to crawl comments and attachments for your selected content types.</td>
</tr>
<tr>
<td>• contentType</td>
<td></td>
</tr>
<tr>
<td>• fieldDefinition</td>
<td></td>
</tr>
<tr>
<td>• isCrawlComments</td>
<td></td>
</tr>
<tr>
<td>• isCrawlFiles</td>
<td></td>
</tr>
<tr>
<td>• isCrawlArticle</td>
<td></td>
</tr>
<tr>
<td>• isCrawlBasicPage</td>
<td></td>
</tr>
<tr>
<td>• isCrawlBasicBlock</td>
<td></td>
</tr>
<tr>
<td>• isCrawlBasicBlock</td>
<td></td>
</tr>
<tr>
<td>• isCrawlCustomContentTypesList</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify DRUPAL as your data source type.</td>
</tr>
<tr>
<td>authType</td>
<td>The type of authentication you are using, whether BASIC-AUTH or OAUTH2.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose</td>
</tr>
<tr>
<td></td>
<td>• FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>true to activate identity crawler. Identity crawler is activated by default. Crawling identity information on users and groups with access to certain documents is useful for user context filtering. Search results are filtered based on the user or their group access to documents. See Identity crawler for more information.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>SecretARN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The Amazon Resource Name (ARN) of a 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:</td>
</tr>
<tr>
<td></td>
<td><strong>If using basic authentication:</strong></td>
</tr>
<tr>
<td></td>
<td>```</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;user name&quot;: &quot;user name&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;password&quot;: &quot;password&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td><strong>If using OAuth 2.0 authentication:</strong></td>
</tr>
<tr>
<td></td>
<td>```</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;Client ID&quot;: &quot;client_id&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;Client secret&quot;: &quot;client_secret&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;user name&quot;: &quot;user name&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;password&quot;: &quot;password&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>Version</td>
<td>The version of this template that is currently supported.</td>
</tr>
</tbody>
</table>

### ACL crawling

When you connect an GitHub data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your GitHub instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

The 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.
For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

## IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the [User Store](#) API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToGetSecret",
      "Effect": "Allow",
      "Action": ["secretsmanager:GetSecretValue"],
      "Resource": ["arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:{{secret_id}}"]
    },
    {
      "Sid": "AllowsAmazonQToDecryptSecret",
      "Effect": "Allow",
      "Action": ["kms:Decrypt"],
      "Resource": [
        "arn:aws:kms:{{region}}:{{account_id}}:key:{{key_id}}"
      ]
    }
  ]
}
```
"arn:aws:kms:{{region}}:{{account_id}}:key/[[key_id]]",
],
"Condition": {
  "StringLike": {
    "kms:ViaService": [
      "secretsmanager.*.amazonaws.com"
    ]
  }
},

{
  "Sid": "AllowsAmazonQToIngestDocuments",
  "Effect": "Allow",
  "Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
  ],
  "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},

{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroups"
  ],
  "Resource": [
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/*"
  ]
},

{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterface",
"ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{account_id}}:network-interface/*",
"Condition": {
"StringLike": {
"aws:RequestTag/AMAZON_Q": "qbusiness_{account_id}_{{application_id}}_*"
},
"ForAllValues:StringEquals": {
"aws:TagKeys": [
"AMAZON_Q"
]
}
}
},
{
"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
"ec2:CreateTags"
],
"Resource": "arn:aws:ec2:{{region}}:{account_id}}:network-interface/*",
"Condition": {
"StringEquals": {
"ec2:CreateAction": "CreateNetworkInterface"
}
}
},
{
"Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterfacePermission"
],
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQServicePrincipal",
            "Effect": "Allow",
            "Principal": {
                "Service": "qbusiness.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "{{source_account}}"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
                }
            }
        }
    ]
}
```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting GitHub (Cloud) to Amazon Q

GitHub is a web-based hosting service for software development providing code storage and management services with version control. You can connect your GitHub instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In GitHub, make sure you have:
• Created a GitHub user with administrative permissions to the GitHub organization.

• Created a personal access token for authentication credentials. See GitHub documentation on creating a personal access token.

• Recommended: Created 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.

• Noted the name of your organization for GitHub the GitHub Enterprise 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.

• Added the following permissions in GitHub:
  • repo:status
  • public_repo
  • repo:invite
  • read:org
  • user:email
  • read:user

In your AWS account, make sure you have:

• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your GitHub authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

![Note]

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.
Using the console

On the GitHub page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Source** – Choose your GitHub source details.
   
   a. **GitHub source** – Choose GitHub Enterprise Cloud.
   
   b. **GitHub host URL** – Enter the GitHub host name with the protocol (http:// or https://). For example: `https://api.github.com`.
   
   c. **GitHub organization name** – You can find your organization name when you log in to GitHub desktop and go to Your organizations under your profile picture dropdown.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.

   **Note**
   
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).

4. **Authentication** – Enter the following information for your AWS Secrets Manager secret.

   a. **Secret name** – A name for your secret.
   
   b. **GitHub token** – Enter the access token you created in GitHub.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.
6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see Identity crawler.

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content. For more information, see IAM role.

8. In **Sync scope**, enter the following information:
   
   a. **Select repositories** – Select between crawling All repositories or crawling Select repositories.
   
   b. **Additional configuration – optional** – Configure the following settings:
      
      • **Content types** – Select the file types you want to include.
      
      • **Regex patterns** – Regular expression patterns to include or exclude certain files. You can add up to 100 patterns.

9. 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 Q for the first time, all content is synced by default.

   • **Full sync** – Sync all content regardless of the previous sync status.
   
   • **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see Sync run schedule.

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See Tags for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.
   
   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.
For more information, see Field mappings.

13. To finish connecting your data source to Amazon Q, select Add data source.

You are taken to the Data source details, where you can view your data source configuration details.

14. In Data source details, choose Sync now to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

Note

You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

Using the API

You use the CreateDataSource action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

GitHub JSON schema

The following is the GitHub JSON schema:

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "connectionConfiguration": {
            "type": "object",
            "properties": {
                "repositoryEndpointMetadata": {
                    "type": "object",
                    "properties": {
                        "type": "object",
                        "properties": {
                            "repositoryEndpointMetadata": {
                                "type": "object",
                                "properties": {
                                    "type": "object",
                                    "properties": {
                                        "type": {
                                            "$ref": "#/properties/connectionConfiguration/properties/repositoryEndpointMetadata/properties/type/properties/"
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```
"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"
                                    ]
                                }
                            }
                        }
                    ]
                }
            }
        }
    }
}
"dataSourceFieldNames": {
  "type": "string"
},
"dateFieldFormat": {
  "type": "string",
  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
},
"required": [
  "indexFieldName",
  "indexFieldType",
  "dataSourceFieldNames"
],
"ghCommit": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "indexFieldName": {
              "type": "string"
            },
            "indexFieldType": {
              "type": "string",
              "enum": [
                "STRING",
                "STRING_LIST",
                "DATE"
              ]
            },
            "dataSourceFieldNames": {
              "type": "string"
            }
          }
        }
      ]
    }
  }
},
"required": ["fieldMappings"]
}
"dateFormat": {  
  "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"
            },
            "dateFormat": {
              "type": "string",
              "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
            }
          }
        }
      ]
    }
  }
}


```json

{}

"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"]
```
"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"
]
"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'"
            }
          }
        }
      ]
    }
  }},
  "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": "string"
            }
        },
        "inclusionFolderNamePatterns": {
            "type": "array",
            "items": {
                "type": "string"
            }
        }
    }
}

GitHub (Cloud)
"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",
  ]
}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
</tbody>
</table>
| hostUrl                     | The GitHub host URL. For example, if you use GitHub Enterprise Cloud: https://api.github
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.com. Or, if you use GitHub Enterprise Server: <a href="https://on-prem-host-url/api/v3/">https://on-prem-host-url/api/v3/</a>.</td>
<td></td>
</tr>
<tr>
<td>organizationName</td>
<td>You can find your organization name when you log in to GitHub desktop and go to <strong>Your organizations</strong> under your profile picture dropdown.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• ghRepository</td>
<td>A list of objects that map the attributes or field names of your Adobe Experience Manager pages and assets to Amazon Q index field names.</td>
</tr>
<tr>
<td>• ghCommit</td>
<td></td>
</tr>
<tr>
<td>• ghIssueDocument</td>
<td></td>
</tr>
<tr>
<td>• ghIssueComment</td>
<td></td>
</tr>
<tr>
<td>• ghIssueAttachment</td>
<td></td>
</tr>
<tr>
<td>• ghPRDocument</td>
<td></td>
</tr>
<tr>
<td>• ghPRComment</td>
<td></td>
</tr>
<tr>
<td>• ghPRAttachment</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>• repositoryFilter</td>
<td>A list of names of the specific repositories you want to index.</td>
</tr>
<tr>
<td>crawlRepository</td>
<td>Specify true to crawl repositories.</td>
</tr>
<tr>
<td>crawlRepositoryDocuments</td>
<td>Specify true to crawl repository documents.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>crawlIssue</td>
<td>Specify true to crawl issues.</td>
</tr>
<tr>
<td>crawlIssueComment</td>
<td>Specify true to crawl issue comments.</td>
</tr>
<tr>
<td>crawlIssueCommentAttachment</td>
<td>Specify true to crawl issue comment attachments.</td>
</tr>
<tr>
<td>crawlPullRequest</td>
<td>Specify true to crawl pull requests.</td>
</tr>
<tr>
<td>crawlPullRequestComment</td>
<td>Specify true to crawl pull request comments.</td>
</tr>
<tr>
<td>crawlPullRequestCommentAttachment</td>
<td>Specify true to crawl pull request comment attachments.</td>
</tr>
<tr>
<td>inclusionFolderNamePatterns</td>
<td>A list of regular expression patterns to include specific 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.</td>
</tr>
<tr>
<td>inclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionFolderNamePatterns</td>
<td>A list of regular expression patterns to exclude specific 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.</td>
</tr>
<tr>
<td>exclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify GITHUB as your data source type.</td>
</tr>
</tbody>
</table>
## Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>enableIdentityCrawler</code></td>
<td>Specify <code>true</code> to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to specific documents.</td>
</tr>
</tbody>
</table>
| `syncMode`             | Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:  
  - Use `FORCED_FULL_CRAWL` to freshly re-crawl all content and replace existing content each time your data source syncs with your index.  
  - Use `FULL_CRAWL` to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index.  
  - Use `CHANGE_LOG` to incrementally crawl only new and modified content each time your data source syncs with your index. |
| `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:  
  ```json  
  {  
    "personalToken": "token"  
  }  
  ``` |
| `version`              | The version of this template that's currently supported. |
ACL crawling

When you connect an GitHub data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your GitHub instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

The 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.

For more information, see:

- Authorization
- Identity crawler
- Understanding User Store

IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToGetSecret",
      "Effect": "Allow"
    }
  ]
}
```
"Action": [
  "secretsmanager:GetSecretValue"
],
"Resource": [
  "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
],
{
  "Sid": "AllowsAmazonQToDecryptSecret",
  "Effect": "Allow",
  "Action": [
    "kms:Decrypt"
  ],
  "Resource": [
    "arn:aws:kms:{{region}}:{{account_id}}:key/[[key_id]]"
  ],
  "Condition": {
    "StringLike": {
      "kms:ViaService": [
        "secretsmanager.*.amazonaws.com"
      ]
    }
  }
},
{
  "Sid": "AllowsAmazonQToIngestDocuments",
  "Effect": "Allow",
  "Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
  ],
  "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},
{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroup"
  ],
"Resource": [  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/*"
],

{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "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]]"
  ]
},

{
  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
  "Effect": "Allow",
  "Action": [  "ec2:CreateNetworkInterface",
               "ec2:DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*",
  "Condition": {  "StringLike": {   "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"  },
   "ForAllValues:StringEquals": {   "aws:TagKeys": [
       "AMAZON_Q"
     ]
  }
  }
},

{
  "Sid": "AllowsAmazonQToCreateTags",
  "Effect": "Allow",
  "Action": [  "ec2:CreateTags"
  ]
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

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

For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting GitHub (Server) to Amazon Q

GitHub is a web-based hosting service for software development providing code storage and management services with version control. You can connect your GitHub instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
# Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

**In GitHub, make sure you have:**

- Created a GitHub user with administrative permissions to the GitHub organization.
- Created a personal access token for authentication credentials. See [GitHub documentation on creating a personal access token](#).
- **Recommended:** Created 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 Server could be `https://on-prem-host-url/api/v3/`.
- Noted the name of your organization for GitHub the GitHub Enterprise 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.
- Added the following permissions in GitHub:
  - repo:status
  - public_repo
  - repo:invite
  - read:org
  - user:email
  - read:user

**In your AWS account, make sure you have:**

- Created an [IAM role](#) for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
• Stored your GitHub authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

**Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

**Using the console**

On the GitHub page, enter the following information:

1. **Name** – Name your data source for easy tracking.

   **Note**: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Source** – Choose your GitHub source details.
   
   a. **GitHub source** – Choose GitHub Enterprise Cloud.
   
   b. **GitHub host URL** – Enter the GitHub host name with the protocol (http:// or https://). For example: `https://on-prem-host-url/api/v3/`.
   
   c. **GitHub organization name** – You can find your organization name when you log in to GitHub desktop and go to **Your organizations** under your profile picture dropdown.
   
   d. **SSL certificate location**— Enter the path to the SSL certificate stored in an Amazon S3 bucket. You use this to connect to github (Server) with a secure SSL connection.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

**Note**

Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).
4. **Authentication** – Enter the following information for your **AWS Secrets Manager secret**.
   
a. **Secret name** – A name for your secret.
   
b. **GitHub token** – Enter the access token you created in GitHub.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:
   
a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   
b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

8. In **Sync scope**, enter the following information:
   
a. **Select repositories** – Select between crawling **All** repositories or crawling **Select repositories**.
   
b. **Additional configuration – optional** – Configure the following settings:
      
      - **Content types** – Select the file types you want to include.
      - **Regex patterns** – Regular expression patterns to include or exclude certain files. You can add up to 100 patterns.

9. 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 Q for the first time, all content is synced by default.

   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.
10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule].

11. **Tags** - **optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags] for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings].

13. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

   ![Note]
   
   You can also choose to view CloudWatch logs for your data source sync job by selecting [View CloudWatch logs]. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

### Using the API

You use the [CreateDataSource] action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.
GitHub JSON schema

The following is the GitHub JSON schema:

```json
{
    "$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"
                            ]
                        }
                    }
                }
            ]
        }
    }
}
"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"
              ]
            }
          }
        }
      ]
    }
  }
}
"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"
              ]
            }
          }
        }
      ]
    }
  }
},
"github": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "indexFieldName": {
              "type": "string"
            },
            "indexFieldType": {
              "type": "string",
              "enum": ["STRING", "STRING_LIST", "DATE"]
            }
          }
        }
      ]
    }
  }
},
"ghPRDocument": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "indexFieldName": {
              "type": "string"
            },
            "indexFieldType": {
              "type": "string",
              "enum": ["STRING", "STRING_LIST", "DATE"]
            }
          }
        }
      ]
    }
  }
},
"github": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "indexFieldName": {
              "type": "string"
            },
            "indexFieldType": {
              "type": "string",
              "enum": ["STRING", "STRING_LIST", "DATE"]
            }
          }
        }
      ]
    }
  }
},
"ghPRDocument": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "indexFieldName": {
              "type": "string"
            },
            "indexFieldType": {
              "type": "string",
              "enum": ["STRING", "STRING_LIST", "DATE"]
            }
          }
        }
      ]
    }
  }
},
"github": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "indexFieldName": {
              "type": "string"
            },
            "indexFieldType": {
              "type": "string",
              "enum": ["STRING", "STRING_LIST", "DATE"]
            }
          }
        }
      ]
    }
  }
},
"ghPRDocument": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "indexFieldName": {
              "type": "string"
            },
            "indexFieldType": {
              "type": "string",
              "enum": ["STRING", "STRING_LIST", "DATE"]
            }
          }
        }
      ]
    }
  }
},
"github": {
  "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"
            }
          }
        }
      ]
    }
  }
}
"dateFormat": {
  "type": "string",
  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
},
"required": [
  "indexFieldName",
  "indexFieldType",
  "dataSourceFieldName"
]
},
"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"
            },
            "dateFormat": {
              "type": "string",
              "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
            }
          }
        }
      ]
    }
  }
}
{  
  "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": "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"
  }
}
"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"
]
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>hostUrl</td>
<td>The GitHub host URL. For example, if you use GitHub Enterprise Cloud: <a href="https://api.github.com">https://api.github.com</a>. Or, if you use GitHub Enterprise Server: <a href="https://on-prem-host-url/api/v3/">https://on-prem-host-url/api/v3/</a>.</td>
</tr>
<tr>
<td>organizationName</td>
<td>You can find your organization name when you log in to GitHub desktop and go to <strong>Your organizations</strong> under your profile picture dropdown.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>ghRepository</td>
<td>A list of objects that map the attributes or field names of your Adobe Experience Manager pages and assets to Amazon Q index field names.</td>
</tr>
<tr>
<td>ghCommit</td>
<td></td>
</tr>
<tr>
<td>ghIssueDocument</td>
<td></td>
</tr>
<tr>
<td>ghIssueComment</td>
<td></td>
</tr>
<tr>
<td>ghIssueAttachment</td>
<td></td>
</tr>
<tr>
<td>ghPRDocument</td>
<td></td>
</tr>
<tr>
<td>ghPRComment</td>
<td></td>
</tr>
<tr>
<td>ghPRAAttachment</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>• repositoryFilter</td>
<td>A list of names of the specific repositories you want to index.</td>
</tr>
<tr>
<td>crawlRepository</td>
<td>Specify true to crawl repositories.</td>
</tr>
<tr>
<td>crawlRepositoryDocuments</td>
<td>Specify true to crawl repository documents.</td>
</tr>
<tr>
<td>crawlIssue</td>
<td>Specify true to crawl issues.</td>
</tr>
<tr>
<td>crawlIssueComment</td>
<td>Specify true to crawl issue comments.</td>
</tr>
<tr>
<td>crawlIssueCommentAttachment</td>
<td>Specify true to crawl issue comment attachments.</td>
</tr>
<tr>
<td>crawlPullRequest</td>
<td>Specify true to crawl pull requests.</td>
</tr>
<tr>
<td>crawlPullRequestComment</td>
<td>Specify true to crawl pull request comments.</td>
</tr>
<tr>
<td>crawlPullRequestCommentAttachment</td>
<td>Specify true to crawl pull request comment attachments.</td>
</tr>
<tr>
<td>• inclusionFolderNamePatterns</td>
<td>A list of regular expression patterns to include specific 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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• exclusionFolderNamePatterns</td>
<td>A list of regular expression patterns to exclude specific 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.</td>
</tr>
<tr>
<td>• exclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• exclusionFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify GITHUB as your data source type.</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>Specify true to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to specific documents.</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>• Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</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 GitHub. The secret must contain a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>`{</td>
</tr>
<tr>
<td></td>
<td>&quot;personalToken&quot;: &quot;token&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>

**ACL crawling**

When you connect an GitHub data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your GitHub instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

The user IDs are mapped as follows:

- `_userid` – 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.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:
• Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.

• Permission to access the User Store API operations to ingest user and group access control information from documents.

• Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.

• Permission to access the SSL certificate stored in your Amazon S3 bucket.

• (Optional) If you’re using Amazon VPC, permission to access your Amazon VPC.

```json
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowsAmazonQToGetS3Objects",
        "Action": [
            "s3:GetObject"
        ],
        "Resource": [
            "arn:aws:s3:::{{input_bucket_name}}/*"
        ],
        "Effect": "Allow",
        "Condition": {
            "StringEquals": {
                "aws:ResourceAccount": "{{account_id}}"
            }"aws:ResourceAccount": "{{account_id}}"
        }},
    },
    {"Sid": "AllowsAmazonQToGetSecret",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:GetSecretValue"
        ],
        "Resource": [
            "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:{{secret_id}}"
        ]},
    {"Sid": "AllowsAmazonQToDecryptSecret",
        "Effect": "Allow",
        "Action": [
```
"kms:Decrypt",
"Resource": [
  "arn:aws:kms:{{region}}:{{account_id}}:key/[[[key_id]]]"
],
"Condition": {
  "StringLike": {
    "kms:ViaService": [
      "secretsmanager.*.amazonaws.com"
    ]
  }
}
},
{
  "Sid": "AllowsAmazonQToIngestDocuments",
  "Effect": "Allow",
  "Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
  ],
  "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},
{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroup"
  ],
  "Resource": [
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**"
  ]
},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
}
"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}}"
]
},
{
    "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*",
    "Condition": {
        "StringLike": {
            "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
        },
        "ForAllValues:StringEquals": {
            "aws:TagKeys": [
                "AMAZON_Q"
            ]
        }
    }
},
{
    "Sid": "AllowsAmazonQToCreateTags",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "CreateNetworkInterface"
        }
    }"}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
    }
  ]
}
```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "{{source_account}}"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
    }
}

For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

**Connecting Gmail to Amazon Q**

Gmail is an 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 connect Gmail instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

**Learn more**

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

**Topics**

- **Prerequisites**
- **Using the console**
- **Using the API**
- **ACL crawling**
- **IAM roles**
Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Gmail, make sure you have:

- Created a Google Cloud Platform admin account and have created a Google Cloud project.
- Activated the 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 about how to create and access your private key, see Create a service account key and Service account credentials on the Google Cloud website.
- Copied your admin account email, your service account email, and your private key to use for authentication.
- 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

In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your Gmail authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note

If you're a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Gmail page, enter the following information:
1. **Name** – Name your data source for easy tracking.

   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   **Note**
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

3. In **Authentication**, for **AWS Secrets Manager secret** – Choose an existing secret or create a Secrets Manager secret to store your Gmail authentication credentials. If you choose to create a secret, an AWS Secrets Manager secret window opens.

   - Enter the following information in the **Create an AWS Secrets Manager secret window**:
     i. **Secret Name** – A name for your secret.
     ii. **Client email** – The client email address that you copied from your Google service account.
     iii. **Admin account email** – The admin account email address that you would like to use.
     iv. **Private key** – The private key that you copied from your Google service account.
     v. Choose **Save**.

4. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).
5. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

6. **In Sync scope**, for **Entity types** – Choose if you want to crawl **Message attachments**. Messages are crawled by default.

7. For **Additional configuration – optional**, enter the following information:

   a. **Date range** – Enter a date range to specify the start and end date of email messages to be crawled.
   
   b. **Email domains** – Include or exclude email messages based on domains.
   
   c. **Keywords in subjects** – Include or exclude email messages based on keywords in their subjects.

   **Note**
   
   You can also choose to include any documents that match all the subject keywords that you have entered.

   d. **Labels** – Add regular expression patterns to include or exclude specific labels. You can add up to 100 patterns.
   
   e. **Attachments** – Add regular expression patterns to include or exclude specific attachments. You can add up to 100 patterns.

8. 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 Q for the first time, all content is synced by default.

   • **Full sync** – Sync all content regardless of the previous sync status.
   
   • **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

9. **In Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

10. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

11. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the [Data source details](#) page after you finish adding your data source. You can choose from two types of fields:
a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

For more information, see [Field mappings](#).

12. To finish connecting your data source to Amazon Q, select **Add data source**.

You are taken to the **Data source details**, where you can view your data source configuration details.

13. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

---

### Note

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

---

### Using the API

You use the **CreateDataSource** action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

### Gmail JSON schema

The following is the Gmail JSON schema:

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "connectionConfiguration": {
            "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": "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"
    ]
  }
}
"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": {
    "isCrawlAcl": {
      "type": "boolean"
    },
    "fieldForUserId": {
      "type": "string"
    },
    "inclusionLabelNamePatterns": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
},
"required": []
}
"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$"
    },
    {
      "type": "string",
      "pattern": ""\n    }
  ]
},
"afterDateFilter": {
  "anyOf": [
    {
      "type": "string",
      "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}Z$"
    },
    {
      "type": "string",
      "pattern": ""\n    }
  ]
},
"isCrawlAttachment": {
  "type": "boolean"
},
"shouldCrawlDraftMessages": {
  "type": "boolean"
},
"required": [
  "isCrawlAttachment",
  "shouldCrawlDraftMessages"
],
"type": {
  "type": "string",
  "pattern": "GMAIL"
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>specific types of content and field mappings. Specify the type of data source and the secret ARN.</td>
</tr>
<tr>
<td>• message</td>
<td>A list of objects that map the attributes or field names of your Gmail messages and attachments to Amazon Q index field names.</td>
</tr>
<tr>
<td>• attachments</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify <code>true</code> to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for <code>UserId</code> for ACL crawling.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>inclusionLabelNamePatterns</td>
<td>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.</td>
</tr>
<tr>
<td>exclusionLabelNamePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionAttachmentTypePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionAttachmentTypePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionAttachmentNamePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionAttachmentNamePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionSubjectFilter</td>
<td></td>
</tr>
<tr>
<td>exclusionSubjectFilter</td>
<td></td>
</tr>
<tr>
<td>inclusionFromFilter</td>
<td></td>
</tr>
<tr>
<td>exclusionFromFilter</td>
<td></td>
</tr>
<tr>
<td>inclusionToFilter</td>
<td></td>
</tr>
<tr>
<td>exclusionToFilter</td>
<td></td>
</tr>
<tr>
<td>inclusionCcFilter</td>
<td></td>
</tr>
<tr>
<td>exclusionCcFilter</td>
<td></td>
</tr>
<tr>
<td>inclusionBccFilter</td>
<td></td>
</tr>
<tr>
<td>exclusionBccFilter</td>
<td></td>
</tr>
</tbody>
</table>

<p>| isSubjectAnd                                         | true to index.                                                                                                                                     |
| 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 automatically crawled.                                               |</p>
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The type of data source. Specify GMAIL as your data source type.</td>
</tr>
<tr>
<td>shouldCrawlDraftMessages</td>
<td>A Boolean value to choose whether you want to crawl draft messages.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose from the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use <code>FORCED_FULL_CRAWL</code> to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use <code>FULL_CRAWL</code> to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
</tbody>
</table>

⚠️ **Important**

Because there is no API to update permanently deleted Gmail messages, a **New, modified, or deleted content sync** does not do the following:

• Remove messages that were permanently deleted from Gmail from your Amazon Q index
• Sync changes in Gmail email labels

To sync your Gmail data source label changes and permanently deleted email messages to your Amazon Q index, you must run full crawls periodically.
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enableIdentityCrawler</td>
<td>Specify true to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to specific documents.</td>
</tr>
<tr>
<td>secretARN</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;adminAccountEmailId&quot;: &quot;${adminAccountEmailId}&quot;   ,</td>
</tr>
<tr>
<td></td>
<td>&quot;clientEmailId&quot;: &quot;${clientId}&quot;   ,</td>
</tr>
<tr>
<td></td>
<td>&quot;privateKey&quot;: &quot;${privateKey}&quot;</td>
</tr>
<tr>
<td>version</td>
<td>The version of the template that's currently supported.</td>
</tr>
</tbody>
</table>

**ACL crawling**

When you connect an Gmail data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Gmail instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

The user IDs are mapped as follows:

- _user_id – User IDs exist in Gmail on files where there are set access permissions. They're mapped from the user emails as the IDs in Gmail.

For more information, see:
To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowsAmazonQToGetSecret",
         "Effect": "Allow",
         "Action": [
            "secretsmanager:GetSecretValue"
         ],
         "Resource": [
            "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
         ]
      },
      {
         "Sid": "AllowsAmazonQToDecryptSecret",
         "Effect": "Allow",
         "Action": [
            "kms:Decrypt"
         ],
         "Resource": [
            "arn:aws:kms:{{region}}:{{account_id}}:key:[[key_id]]"
         ]
      }
   ]
}
```
"Condition": {
    "StringLike": {
        "kms:ViaService": [
            "secretsmanager.*.amazonaws.com"
        ]
    }
},

{
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ],
    "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},

{
    "Sid": "AllowsAmazonQToIngestPrincipalMapping",
    "Effect": "Allow",
    "Action": [
        "qbusiness:PutGroup",
        "qbusiness:CreateUser",
        "qbusiness:DeleteGroup",
        "qbusiness:UpdateUser",
        "qbusiness:ListGroups"
    ],
    "Resource": [
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**"
    ]
},

{
    "Sid": "AllowsAmazonQToCreateAndDeleteNI",
    "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]]"
],
}
{
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterface",
"ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringLike": {
"aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
},
"ForAllValues:StringEquals": {
"aws:TagKeys": [
"AMAZON_Q"
]
}
}
},
{
"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
"ec2:CreateTags"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringEquals": {
"ec2:CreateAction": "CreateNetworkInterface"
}
}
},
{
"Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterfacePermission"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQServicePrincipal",
            "Effect": "Allow",
            "Principal": {
                "Service": "qbusiness.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "{{source_account}}"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
                }
            }
        }
    ]
}
```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting Google Drive to Amazon Q

Google Drive is a cloud-based file storage service. Amazon Q can connect to your Google Drive instances. You can connect Google Drive instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- User context filtering
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

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.

• 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 Q 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 Q 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.

  **To support the Forms API, add the following additional scope:**
  • https://www.googleapis.com/auth/forms.body.readonly
In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your Google Drive authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

**Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

### Using the console

On the GoogleDrive page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   **Note**

   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

3. For **Authentication** – Choose between Google service account and OAuth 2.0 authentication, based on your use case.

4. **AWS Secrets Manager secret** – Choose an existing secret or create a Secrets Manager secret to store your GoogleDrive authentication credentials. If you choose to create a secret, an AWS Secrets Manager secret window opens.
• If you choose **Existing**, select an existing secret for **Select secret**.

If you choose **New**, enter the following information in the **New AWS Secrets Manager secret** section:

i. **Secret name** – A name for your secret.

ii. If you chose **Google service account**, enter the **Secret Name**, **Admin account email**, **Client email**, and **Private Key** that you created in your service account. Then, choose **Save and add secret**.

iii. If you chose **OAuth 2.0 authentication**, enter the details of **Secret Name**, **Client ID**, **Client secret** and **Refresh token** that you created in your service account. Then, choose **Save and add secret**.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

8. In **Sync scope**, for **Sync contents** – Select from the following choices to index **My Drive & Shared with me, Shared drives**, and **Comments**. Files are crawled by default.

9. In **Additional configuration - optional**, enter the following optional information:

   a. **Maximum file size** – Set the maximum file size value that Amazon Q will crawl.

   b. **User email** – Add the user email IDs that you want to include or exclude.

   c. **Shared drives** – Add the shared drives that you want to include or exclude.
d. **Mime types** – Add the MIME types that you want to include or exclude.

e. **Attachment regex patterns** – Add the regular expression patterns to include or exclude certain attachments for all supported entities. You can add up to 100 patterns.

10. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New or modified content sync** – Sync only new and modified documents.
   - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

   For more details, see [Sync mode](#).

11. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

12. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

13. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

14. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

15. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.
Note
You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

Using the API

You use the CreateDataSource action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

Google Drive JSON schema

The following is the Google Drive JSON schema:

```json
{
    "$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'"
            }
          }
        },
        {
          "indexFieldName",
          "indexFieldType",
          "dataSourceFieldName"
        }
      ]
    }
  }
},
"required": [
  "indexFieldName",
  "indexFieldType",
  "dataSourceFieldName"
]
"fieldMappings"
 ]
}
},
"additionalProperties": {
 "type": "object",
 "properties": {
 "maxFileSizeInMegaBytes": {
 "type": "string"
 },
 "isCrawlComment": {
 "type": "boolean"
 },
 "isCrawlMyDriveAndSharedWithMe": {
 "type": "boolean"
 },
 "isCrawlSharedDrives": {
 "type": "boolean"
 },
 "isCrawlAcl": {
 "type": "boolean"
 },
 "fieldForUserId": {
 "type": "string"
 },
 "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"
    }
}
"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"
      }
    ]
  }
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source. This data source doesn't 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.</td>
</tr>
<tr>
<td>authType</td>
<td>Choose between serviceAccount and OAuth2, based on your use case.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• file</td>
<td>A list of objects that map the attributes or field names of your Google Drive to Amazon Q index field names.</td>
</tr>
<tr>
<td>• comment</td>
<td>A list of objects that map the attributes or field names of your Google Drive to Amazon Q index field names.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>• maxFileSizeInMegaBytes</td>
<td>Enter the maximum file size value in MBs that Amazon Q should crawl and index from your Google Drive data source.</td>
</tr>
<tr>
<td>• iscrawlComment</td>
<td>true to index comments in your Google Drive data source.</td>
</tr>
<tr>
<td>• iscrawlMyDriveAndSharedWithMe</td>
<td>true to index MyDrive and Shared With Me Drives in your Google Drive data source.</td>
</tr>
<tr>
<td>• iscrawlSharedDrives</td>
<td>true to index Shared Drives in your Google Drive data source.</td>
</tr>
<tr>
<td>• excludeUserAccounts</td>
<td>A list of regular expression patterns to exclude specific files in your Google Drive data source.</td>
</tr>
<tr>
<td>• excludeSharedDrives</td>
<td>Files that match the patterns are excluded from 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.</td>
</tr>
<tr>
<td>• excludeMimeTypes</td>
<td>A list of regular expression patterns to include specific files in your Google Drive data source.</td>
</tr>
<tr>
<td>• exclusionFileTypePatterns</td>
<td>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.</td>
</tr>
<tr>
<td>• exclusionFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• exclusionFilePathFilter</td>
<td></td>
</tr>
<tr>
<td>• includeUserAccounts</td>
<td>A list of regular expression patterns to include specific files in your Google Drive data source.</td>
</tr>
<tr>
<td>• includeSharedDrives</td>
<td>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.</td>
</tr>
<tr>
<td>• includeMimeTypes</td>
<td></td>
</tr>
<tr>
<td>• inclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionFilePathFilter</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify GOOGLEDRIVEV2 as your data source type.</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>True to activate identity crawler. Identity crawler is activated by default. Crawling identity information on users and groups with access to certain documents is useful for user context filtering. Search results are filtered based on the user or their group access to documents. See <a href="#">Identity crawler</a> for more information.</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 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:  

```json
{
   "clientEmail": "user account email",
   "adminAccountEmail": "service account email",
   "privateKey": "private key"
}
```

If using OAuth 2.0 authentication:  

```json
{
   "clientID": "OAuth client ID",
   "clientSecret": "client secret",
   "refreshToken": "refresh token"
}
```

| version       | The version of this template that's currently supported.                                                                                     |

**User context filtering**

When you connect an GoogleDrive data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your GoogleDrive instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

The GoogleDrive group and user IDs are mapped as follows:
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 username is mapped to the _user_id field.

For more information, see:

- Authorization
- Identity crawler
- Understanding User Store

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowsAmazonQToGetSecret",
         "Effect": "Allow",
         "Action": [
            "secretsmanager:GetSecretValue"
         ],
         "Resource": [
            "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
         ]
      },
      {
         "Sid": "AllowsAmazonQToDecryptSecret",
```
"Effect": "Allow",
"Action": [
  "kms:Decrypt"
],
"Resource": [
  "arn:aws:kms:{{region}}:{{account_id}}:key/[{{key_id}}]"
],
"Condition": {
  "StringLike":{
    "kms:ViaService": ["secretsmanager.*.amazonaws.com"
  }
}
},
{
  "Sid": "AllowsAmazonQToIngestDocuments",
  "Effect": "Allow",
  "Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
  ],
  "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},
{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroups"
  ],
  "Resource": [
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/*"
  ],
}
"Sid": "AllowsAmazonQToCreateAndDeleteNI",
"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]
]
},
{
    "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
    "Condition": {
        "StringLike": {
            "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
        },
        "ForAllValues/StringEquals": {
            "aws:TagKeys": [
                "AMAZON_Q"
            ]
        }
    }
},
{
    "Sid": "AllowsAmazonQToCreateTags",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "CreateNetworkInterface"
        }
    }
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
        }
      }
    }
  ]
}
```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

**Connecting IBM DB2 to Amazon Q**

IBM DB2 is a relational database management system developed by IBM. If you are an AWS user, you can use Amazon Q to index your IBM DB2 data source.

You can connect your IBM DB2 instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

The Amazon Q IBM DB2 data source connector supports DB2 11.5.7.

**Learn more**

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

**Topics**

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles
Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In IBM DB2, make sure you have:

• Noted your database user name and password.

  Important
  As a best practice, provide Amazon Q with read-only database credentials.

• Copied your database host URL, port, and instance.

In your AWS account, make sure you have:

• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your IBM DB2 authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

  Note
  If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the IBM DB2 page, enter the following information:

1. Name – Name your data source for easy tracking.

   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In Source, enter the following information:

   a. Host – Enter the database host name.
b. **Port** – Enter the database port.

c. **Instance** – Enter the database instance.

d. **Enable SSL certificate location** – Choose to enter the Amazon S3 path to your SSL certificate file.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See **Authorization** for more details.

    🔄 **Note**

    Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see **Integrating with an Identity Provider (IdP)**.

4. In **Authentication** – Enter the following information for your **AWS Secrets Manager secret**.

   a. **Secret name** – A name for your secret.

   b. For **Database user name**, and **Password** – Enter the authentication credential values you copied from your database.

   c. Choose **Save**.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

      For more information, see **VPC**.

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

      For more information, see **IAM role**.

7. In **Sync scope**, enter the following information:
• **SQL query** – Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Q 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 names of the document body column within your database table.

8. In **Additional configuration – optional** – Configure the following settings:

• **Change-detecting columns** – Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q 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 Q 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.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

• **Full sync** – Sync all content regardless of the previous sync status.

• **New or modified content sync** – Sync only new and modified documents.

• **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.
For more details, see Sync mode.

10. In Sync run schedule, for Frequency – Choose how often Amazon Q will sync with your data source. For more details, see Sync run schedule.

11. Tags - optional – Add tags to search and filter your resources or track your AWS costs. See Tags for more details.

12. Field mappings – A list of data source document attributes to map to your index fields. Add the fields from the Data source details page after you finish adding your data source. You can choose from two types of fields:
   a. Default – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.
   b. Custom – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see Field mappings.

13. To finish connecting your data source to Amazon Q, select Add data source.

   You are taken to the Data source details, where you can view your data source configuration details.

14. In Data source details, choose Sync now to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

   Note

   You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

Using the API

You use the CreateDataSource action to connect a data source to your Amazon Q application.
Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**JSON schema**

The following is the JSON schema:

```json
{
  "$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": [
    "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"
        },
        "serverlessAurora": {
            "type": "string"
        }
    }
}
The following table provides information about important JSON keys to configure.
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>Required configuration information for connecting your data source.</td>
</tr>
<tr>
<td></td>
<td>• <code>dbType</code>—The type of Java database you are using, whether <code>mysql</code>, <code>db2</code>, <code>postgresql</code>, <code>oracle</code>, or <code>sqlserver</code>.</td>
</tr>
<tr>
<td></td>
<td>• <code>dbHost</code>—The database host name.</td>
</tr>
<tr>
<td></td>
<td>• <code>dbPort</code>—The database port.</td>
</tr>
<tr>
<td></td>
<td>• <code>dbInstance</code>—The database instance.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source.</td>
</tr>
<tr>
<td></td>
<td>For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td></td>
<td>Specify the type of data source and the secret ARN.</td>
</tr>
<tr>
<td>document</td>
<td>A list of objects that map the attributes or field names of your database</td>
</tr>
<tr>
<td></td>
<td>content to Amazon Q index field names. For more information, see <strong>Mapping</strong></td>
</tr>
<tr>
<td></td>
<td><strong>data source fields</strong></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td></td>
<td>Use to include or exclude specific content in your database data source.</td>
</tr>
<tr>
<td>primaryKey</td>
<td>Provide the primary key for the database table. This identifies a table</td>
</tr>
<tr>
<td></td>
<td>within your database.</td>
</tr>
<tr>
<td>titleColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>bodyColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>sqlQuery</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>timestampColumn</td>
<td>Enter the name of the column which contains time stamps. Amazon Q uses time stamp information to detect changes in your content and sync only changed content.</td>
</tr>
<tr>
<td>timestampFormat</td>
<td>Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.</td>
</tr>
<tr>
<td>timezone</td>
<td>Enter the name of the column which contains time zones for the content to be crawled.</td>
</tr>
<tr>
<td>changeDetectingColumns</td>
<td>Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q will re-index content when there is a change in any of these columns</td>
</tr>
<tr>
<td>allowedUsersColumns</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>allowedGroupsColumn</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>sourceURIColumn</td>
<td>Enter the name of the column which contains Source URLs to be indexed.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>isSslEnabled</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify JDBC as your data source type.</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose</td>
</tr>
<tr>
<td></td>
<td>• FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index</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:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;user name&quot;: &quot;database user name&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;password&quot;: &quot;password&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>The version of the template that is currently supported.</td>
</tr>
</tbody>
</table>

### ACL crawling

When you connect a database data source to Amazon Q, Amazon Q crawls user and group information from a column in the source table. You specify this column in the console or using the configuration parameter as part of the `CreateDataSource` operation.

If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

### IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the `BatchPutDocument` and `BatchDeleteDocument` operations to ingest documents.
- Permission to access the [User Store](#) API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- Permission to access the SSL certificate stored in your Amazon S3 bucket.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToGetS3Objects",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::{{input_bucket_name}}/*"
      ],
      "Effect": "Allow",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "{{account_id}}"
        }
      }
    },
    {
      "Sid": "AllowsAmazonQToGetSecret",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
      ]
    },
    {
      "Sid": "AllowsAmazonQToDecryptSecret",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:{{region}}:{{account_id}}:key:[[key_id]]"
      ]
    }
  ]
}
```
"Condition": {
    "StringLike": {
        "kms:ViaService": [
            "secretsmanager.*.amazonaws.com"
        ]
    }
},

{  
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ],
    "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},

{  
    "Sid": "AllowsAmazonQToIngestPrincipalMapping",
    "Effect": "Allow",
    "Action": [
        "qbusiness:PutGroup",
        "qbusiness:CreateUser",
        "qbusiness:DeleteGroup",
        "qbusiness:UpdateUser",
        "qbusiness:ListGroups"
    ],
    "Resource": [
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**
    ]
},

{  
    "Sid": "AllowsAmazonQToCreateAndDeleteNI",
    "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]]"
],

{  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
  "Effect": "Allow",
  "Action": [  "ec2:CreateNetworkInterface",
              "ec2:DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "AMAZON_Q"
      ]
    }
  }
},

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

{  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [  "ec2:CreateNetworkInterfacePermission"
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
            "Effect": "Allow",
            "Principal": {
                "Service": "qbusiness.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "{{source_account}}"
                },
                "ArnLike": {

```
"aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
]
}
]
}

For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

## Connecting Jira to Amazon Q

Jira is a project management tool for software development, product management, and bug tracking. You can connect your Jira instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

### Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

### Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

### Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

**In Jira, make sure you have:**
• Created Jira API token authentication credentials that include a Jira ID (user name or email) and a Jira credential (Jira API token). See Atlassian documentation on managing API tokens.

• Noted the Jira account URL from your Jira account settings. For example, company.atlassian.net.

In your AWS account, make sure you have:

• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your Jira authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Jira page, enter the following information:

1. Name – Name your data source for easy tracking.

   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. Source – Enter your Jira URL. For example: https://www.jira.com/.

3. Authorization – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.
4. Authentication – Enter the following information for your AWS Secrets Manager secret.
   a. Secret name – A name for your secret.
   b. Jira ID – Your Jira ID.
   c. Password/Token – Your Jira token.

5. Configure VPC and security group – optional – Choose whether you want to use a VPC. If you do, enter the following information:
   a. Subnets – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   b. VPC security groups – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see VPC.

6. Identity crawler – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see Identity crawler.

7. IAM role – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see IAM role.

8. In Sync scope, enter the following information:
   a. Choose to either sync All projects or sync Only specific projects. If you choose to sync Only specific projects, enter the Jira Project Key ID.
   b. Additional configuration – optional – Choose from the following options to limit the scope for content to be indexed. Otherwise, all content will be synced by default.
      - Statuses – Add status values to index.
- **Additional elements** – Choose whether to index **Comments, Attachments, or Worklogs**.
- **Issue types** – Choose the issues types you want to index.
- **Regex patterns** – Add regex patterns to include or exclude file names, file types, or file paths. You can have a total of 100 patterns.

9. **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 Q for the first time, all content is synced by default.

   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the [Data source details](#) page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the [Data source details](#), where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.
**Note**

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**Jira JSON schema**

The following is the Jira JSON schema:

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "connectionConfiguration": {
            "type": "object",
            "properties": {
                "repositoryEndpointMetadata": {
                    "type": "object",
                    "properties": {
                        "jiraAccountUrl": {
                            "type": "string",
                            "pattern": "https://.*"
                        }
                    },
                    "required": ["jiraAccountUrl"]
                }
            },
            "required": ["repositoryEndpointMetadata"]
        }
    }
}```
"repositoryConfigurations": {
  "type": "object",
  "properties": {
    "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"
  ]
},
"Jira": 591
[{
  "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'"
            }
          }
        }
      ]
    }
  }
}]

"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"
]
],
"issue": {
"Jira": 592
}
"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"
],
"project": {
    "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"
]}
]
"worklog": {
    "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"
]
}
"additionalProperties": {
  "type": "object",
  "properties": {
    "isCrawlAcl": {
      "type": "boolean"
    },
    "fieldForUserId": {
      "type": "string"
    }
  }
}
},
  "issuetype": {
    "type": "array",
    "items": {
      "type": "string",
      "enum": [
        "Bug",
        "Story",
        "Epic",
        "Task"
      ]
    }
  },
  "status": {
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "project": {
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "issueSubEntityFilter": {
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "inclusionPatterns": {
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "exclusionPatterns": {
    "type": "array",
    "items": {
      "type": "string"
    }
  }
},
Jira
596
The following table provides information about important JSON keys to configure.
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>jiraAccountUrl</td>
<td>Enter the Jira account URL from your Jira account settings.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• attachment</td>
<td>A list of objects that map the attributes or field names of your Jira pages and assets to Amazon Q index field names.</td>
</tr>
<tr>
<td>• comment</td>
<td></td>
</tr>
<tr>
<td>• issue</td>
<td></td>
</tr>
<tr>
<td>• project</td>
<td></td>
</tr>
<tr>
<td>• worklog</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>• issuetype</td>
<td>Choose to customize the scope of your crawl with specific entities. You can add specific status types, additional elements, and issue types to crawl.</td>
</tr>
<tr>
<td>• status</td>
<td></td>
</tr>
<tr>
<td>• project</td>
<td></td>
</tr>
<tr>
<td>• issueSubEntityFilter</td>
<td></td>
</tr>
<tr>
<td>• inclusionPatterns</td>
<td>A list of regular expression patterns to include specific content in your Jira data source. Content that matches the patterns</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>are included in the index. Contents that doesn't match the pattern 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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>exclusionPatterns</strong></td>
</tr>
<tr>
<td></td>
<td>A list of regular expression patterns to exclude specific content in your Jira 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.</td>
</tr>
<tr>
<td><strong>type</strong></td>
<td>The type of data source. Specify JIRA as your data source type.</td>
</tr>
<tr>
<td><strong>enableIdentityCrawler</strong></td>
<td>Specify <code>true</code> to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to specific documents.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use <strong>FORCED_FULL_CRAWL</strong> to freshly re-crawl all content and replace existing content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>• Use <strong>FULL_CRAWL</strong> to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>• Use <strong>CHANGE_LOG</strong> to incrementally crawl only new and modified content each time your data source syncs with your index.</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 Jira. The secret must contain a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>```json</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;Jira ID&quot;: &quot;Jira user name or email host URL&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;Password/Token&quot;: &quot;Jira API token&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>
ACL crawling

When you connect an Jira data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Jira instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the `BatchPutDocument` and `BatchDeleteDocument` operations to ingest documents.
- Permission to access the [User Store](#) API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- *(Optional)* If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [  
    {  
      "Sid": "AllowsAmazonQToGetSecret",
      "Effect": "Allow",
```
"Action": [
    "secretsmanager:GetSecretValue"
],
"Resource": [
    "arn:aws:secretsmanager:{region}:{account_id}:secret:{{secret_id}}"
],

{
    "Sid": "AllowsAmazonQToDecryptSecret",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:{region}:{account_id}:key/[[key_id]]"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
                "secretsmanager.*.amazonaws.com"
            ]
        }
    }
},

{
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ],
    "Resource": "arn:aws:qbusiness:{region}:{source_account}:application/{application_id}/index/{{index_id}}"
},

{
    "Sid": "AllowsAmazonQToIngestPrincipalMapping",
    "Effect": "Allow",
    "Action": [
        "qbusiness:PutGroup",
        "qbusiness:CreateUser",
        "qbusiness:DeleteGroup",
        "qbusiness:UpdateUser",
        "qbusiness:ListGroup"
    ],
}
"Resource": [  
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",  
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",  
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/*"
],

{  
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",  
  "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]]"
  ]
},

{  
  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",  
  "Effect": "Allow",  
  "Action": [  
    "ec2:CreateNetworkInterface",  
    "ec2:DeleteNetworkInterface"
  ],  
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*",  
  "Condition": {  
    "StringLike": {  
      "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    },  
    "ForAllValues:StringEquals": {  
      "aws:TagKeys": [  
        "AMAZON_Q"
      ]
    }
  }
},

{  
  "Sid": "AllowsAmazonQToCreateTags",  
  "Effect": "Allow",  
  "Action": [  
    "ec2:CreateTags"
  ]
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
      "Effect": "Allow",
      "Action": ["ec2:CreateNetworkInterfacePermission"],
      "Resource": "arn:aws:ec2:{region}:{account_id}:network-interface/**",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/AMAZON_Q": "qbusiness_{account_id}_{application_id}_*"
        }
      }
    },
    {
      "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:DescribeVpcs",
        "ec2:DescribeRegions",
        "ec2:DescribeNetworkInterfacePermissions",
        "ec2:DescribeSubnets"
      ],
      "Resource": "*"
    }
  ]
}
```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting Microsoft Exchange to Amazon Q

Microsoft Exchange is an enterprise collaboration tool for messaging, meetings, and file sharing. You can connect Microsoft Exchange instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Microsoft Exchange, make sure you have:

- Created a Microsoft Exchange account in Office 365.
- Copied your Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal. For more information, see Find your Microsoft 365 tenant ID on the Microsoft website.
- Configured an OAuth 2.0 credential token containing a client ID and client secret.
- Added the following permissions for the connector application:

<table>
<thead>
<tr>
<th>Microsoft Graph</th>
<th>Office 365 Exchange Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mail.Read (Application)</td>
<td>full_access_as_app (Application)</td>
</tr>
<tr>
<td>Mail.ReadBasic (Application)</td>
<td></td>
</tr>
<tr>
<td>Mail.ReadBasic.All (Application)</td>
<td></td>
</tr>
<tr>
<td>Calendars.Read (Application)</td>
<td></td>
</tr>
<tr>
<td>User.Read.All Application)</td>
<td></td>
</tr>
<tr>
<td>Contacts.Read (Application)</td>
<td></td>
</tr>
<tr>
<td>Notes.Read.All (Application)</td>
<td></td>
</tr>
<tr>
<td>Directory.Read.All (Application)</td>
<td></td>
</tr>
<tr>
<td>EWS.AccessAsUser.All (Delegated)</td>
<td></td>
</tr>
</tbody>
</table>

In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
• Stored your Microsoft Exchange authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Microsoft Exchange page, enter the following information:

1. **Name** – Name your data source for easy tracking.

   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In Source, enter the following information:

   • **Tenant ID** – Enter your tenant id. Your Microsoft tenant ID is a globally unique identifier that's necessary to configure each connector instance. Your tenant ID is different from your organization name or domain and can be found in the properties section of your Microsoft account dashboard.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.

   Note
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).

4. **Authentication** – Choose between New and Existing.

   • If you choose Existing, select an existing secret for Select secret.
If you choose **New**, enter the following information in the **New AWS Secrets Manager secret** section:

i. **Secret name** – A name for your secret.

ii. For **Client ID, Client secret** – Enter the authentication credential values that you generated from your Exchange account.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

7. In **Sync scope**, choose from the following options:

   - **UserIDs** – Select to filter content by specific user email IDs.
   - **User email ID** – Upload a file with user email ids to filter content by. Email IDs must be formatted on a separate line in the file.

8. For **Additional configuration – optional**, choose from the following options:

   - **Entity types** – Choose whether you want to crawl the following entities: **Calendar**, **OneNotes**, and **Contacts**.
   - **Calendar crawling** – Enter the date range for which the connector will crawl your calendar content.
   - **Include email** – Enter the email from domains, email to domains, and subjects you wish to include or exclude in your application.
   - **Shared folders access** – Enable ACL crawling for shared folders.
• **Regex for domains** – Add patterns to include and exclude certain email domains from your application.

• **Regex patterns** – Add regular expression patterns to include or exclude certain files. You can add up to 100 patterns.

9. **In Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   • **Full sync** – Sync all content regardless of the previous sync status.
   
   • **New or modified content sync** – Sync only new and modified documents.
   
   • **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

   For more details, see [Sync mode](#).

10. **In Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the [Data source details](#) page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.

    You are taken to the [Data source details](#), where you can view your data source configuration details.

14. **In Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.
Note

You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

Using the API

You use the CreateDataSource action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

Microsoft Exchange JSON schema

The following is the Microsoft Exchange JSON schema:

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "connectionConfiguration": {
            "type": "object",
            "properties": {
                "repositoryEndpointMetadata": {
                    "type": "object",
                    "properties": {
                        "tenantId": {
                            "type": "string",
                            "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$",
                            "minLength": 36,
                            "maxLength": 36
                        }
                    }
                }
            }
        }
    }
}
```

Microsoft Exchange
"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": "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"
          ]
        }
      }
    }
  }
}
"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"
            }
          },
          "required": ["indexFieldName"
        ]
      ]
    }
  }
}
"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"
}
}
]
}
}
},
"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"
}
}
]
}
},
"required": [
"indexFieldName",
"indexFieldType",
"dataSourceFieldName"
]
},
"required":[
"fieldMappings"
]
}


```json
{
  "fieldMappings": [
    {
      "indexFieldName": "string",
      "indexFieldType": "STRING",
      "dataSourceFieldName": "string",
      "dateFieldFormat": "yyyy-MM-dd'T'HH:mm:ss'Z'"
    }
  ],
  "required": [
    "indexFieldName",
    "indexFieldType",
    "dataSourceFieldName"
  ]
}
}
"notes": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "indexFieldName": "string",
            "indexFieldType": "string",
            "enum": ["STRING", "DATE"],
            "dataSourceFieldName": "string",
            "dateFieldFormat": "yyyy-MM-dd'T'HH:mm:ss'Z'"
          }
        }
      ],
      "required": [
    "indexFieldName",
    "indexFieldType",
    "dataSourceFieldName",
    "dateFieldFormat"
  ]
}
}
```

"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"
    ]
}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>tenantId</td>
<td>The Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal.</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• email</td>
<td>A list of objects that map the attributes or field names of your Microsoft Exchange data source.</td>
</tr>
<tr>
<td>• attachment</td>
<td></td>
</tr>
<tr>
<td>• calendar</td>
<td></td>
</tr>
<tr>
<td>• contacts</td>
<td></td>
</tr>
<tr>
<td>• notes</td>
<td></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 Exchange data source. This includes your client ID and your client secret.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for content in your data source</td>
</tr>
<tr>
<td>inclusionPatterns</td>
<td>A list of regular expression patterns to <em>include</em> specific files in your Exchange 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.</td>
</tr>
<tr>
<td>• inclusionUsersList</td>
<td></td>
</tr>
<tr>
<td>• inclusionUsersFileName</td>
<td></td>
</tr>
<tr>
<td>• inclusionDomainUsers</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>exclusionPatterns</td>
<td>A list of regular expression patterns to <em>exclude</em> specific files in your 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.</td>
</tr>
<tr>
<td>exclusionUsersList</td>
<td></td>
</tr>
<tr>
<td>exclusionUsersFileName</td>
<td></td>
</tr>
<tr>
<td>exclusionDomainUsers</td>
<td></td>
</tr>
<tr>
<td>startCalendarDateTime</td>
<td>Use to specify the date and time for Calendar content to be crawled by Amazon Q.</td>
</tr>
<tr>
<td>endCalendarDateTime</td>
<td>Use to specify the date and time for Calendar content to be crawled by Amazon Q.</td>
</tr>
<tr>
<td>subject</td>
<td>Use to specify email subject lines to be crawled.</td>
</tr>
<tr>
<td>emailFrom</td>
<td>Use to specify emails to be crawled based on sender.</td>
</tr>
<tr>
<td>emailTo</td>
<td>Use to specify emails to be crawled based on recipient.</td>
</tr>
<tr>
<td>- crawlCalendar</td>
<td>true to index this content in your Microsoft Exchange data source.</td>
</tr>
<tr>
<td>- crawlNotes</td>
<td></td>
</tr>
<tr>
<td>- crawlContacts</td>
<td></td>
</tr>
<tr>
<td>- crawlFolderAcl</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify MSEXCHANGE as your data source type.</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>true to activate identity crawler. Identity crawler is activated by default. Crawling identity information on users and groups with access to specific documents is useful for user context filtering. Search results are filtered based on the user or their group access to documents. See Identity crawler for more information.</td>
</tr>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>
**ACL crawling**

When you connect an Exchange data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Exchange instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

The Exchange group and user IDs are mapped as follows:

- `_tenant_id` – Your Microsoft tenant ID is a globally unique identifier that's necessary to configure each connector instance. Your tenant ID is different from your organization name or domain and can be found in the properties section of your Microsoft account dashboard.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the `BatchPutDocument` and `BatchDeleteDocument` operations to ingest documents.
- Permission to access the [User Store](#) API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToGetSecret",
```
"Effect": "Allow",
"Action": [
  "secretsmanager:GetSecretValue"
],
"Resource": [
  "arn:aws:secretsmanager:{region}:{account_id}:secret:[[secret_id]]"
],

{
  "Sid": "AllowsAmazonQToDecryptSecret",
  "Effect": "Allow",
  "Action": [
    "kms:Decrypt"
  ],
  "Resource": [
    "arn:aws:kms:{region}:{account_id}:key:[[key_id]]"
  ],
  "Condition": { 
    "StringLike": { 
      "kms:ViaService": [ 
        "secretsmanager.*.amazonaws.com"
      ] 
    }
  }
},

{
  "Sid": "AllowsAmazonQToIngestDocuments",
  "Effect": "Allow",
  "Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
  ],
  "Resource": "arn:aws:qbusiness:{region}:{source_account}:application/{{application_id}}/index/{{index_id}}"
},

{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroup"
  ]
}
"Resource": [
   "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
   "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
   "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**"
],

"Sid": "AllowsAmazonQToCreateAndDeleteNI",
"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]]"
]
},

"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
   "ec2:CreateNetworkInterface",
   "ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
   "StringLike": {
      "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
   },
   "ForAllValues:StringEquals": {
      "aws:TagKeys": [
         "AMAZON_Q"
      ]
   }
}
},

"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission"
            ],
            "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateNetworkInterface"
                }
            }
        },
        {
            "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeNetworkInterfaceAttribute",
                "ec2:DescribeVpcs",
                "ec2:DescribeRegions",
                "ec2:DescribeNetworkInterfacePermissions",
                "ec2:DescribeSubnets"
            ],
            "Resource": "*"
        }
    ]
}
```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting Microsoft OneDrive to Amazon Q

Microsoft OneDrive is a cloud-based storage service that you can use to store, share, and host your content. You can connect Microsoft OneDrive instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

**In your Azure Active Directory (AD) application, 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 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)

**In your AWS account, make sure you have:**

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your Microsoft OneDrive authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

ℹ️ **Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.
For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

### Using the console

On the **OneDrive** page, enter the following information:

1. **In Source**, enter the following information:
   
   - **OneDrive Tenant ID** Enter your OneDrive Tenant ID without the protocol. You can find your OneDrive Tenant ID under Directory ID in the Microsoft Azure AD admin center.

2. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   ![Note](#)
   
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

3. **In Authentication** – Choose between New and Existing.
   
   - If you choose **Existing**, select an existing secret for Select secret.
     
     If you choose **New**, enter the following information in the **New AWS Secrets Manager secret** section:
     
     i. **Secret name** – A name for your secret.
     
     ii. For **Application ID** and **Application password** – Enter the authentication credential values from your OneDrive account and then choose Save authentication.

4. **Configure VPC and security group** – optional – Choose whether you want to use a VPC. If you do, enter the following information:
   
   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.
For more information, see VPC.

5. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see Identity crawler.

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.
   For more information, see IAM role.

7. In **Sync scope**, for **Select OneDrive users**, choose between the following options:
   - **Add a user name file** – Choose to add a user names file saved in an Amazon S3 bucket. Provide the path to the file by choosing Browse.

   **Note**
   If you choose this option, the IAM role for the data source must have read permissions for the Amazon S3 bucket where the file is stored.

   - **Add user names here** – You can add a maximum of 10 users using this option. To add more than 10 users, please create a file containing the user names and choose Add a user name file.

8. For **Additional configuration** – optional:
   - For **Regex for OneNote** and **Regex Patterns** – Add regular expression patterns to include or exclude certain files. You can add up to 100 patterns.

9. 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 Q for the first time, all content is synced by default.
   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New, modified, or deleted content sync** – Only sync new, modified, and deleted content.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see Sync run schedule.

11. **Tags** – optional – Add tags to search and filter your resources or track your AWS costs. See Tags for more details.
12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

   ![Note](https://example.com)

   You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

---

**Using the API**

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**Microsoft OneDrive JSON schema**

The following is a the Microsoft OneDrive JSON schema:

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
...
```
"type": "object",
"properties": {
  "connectionConfiguration": {
    "type": "object",
    "properties": {
      "repositoryEndpointMetadata": {
        "type": "object",
        "properties": {
          "tenantId": {
            "type": "string",
            "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$",
            "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"
                },
                "indexFieldTyp": {
                  "type": "string",
                  "enum": [
                    "STRING",
                    "STRING_LIST",
                    "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": { 
 "isCrawlAcl": { 
 "type": "boolean"
 },
 "fieldForUserId": { 
 "type": "string"
 },
 "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"
  }
}
}
"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
}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source. This includes the tenant ID in the form of the OneDrive site URL.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>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.</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 OneDrive. The secret must contain a JSON structure with the following keys:</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;clientID&quot;: &quot;OAuth Client ID&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;password&quot;: &quot;client secret&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• userNameFilter</td>
<td>A collection of strings that specifies which entities to filter.</td>
</tr>
<tr>
<td>• userFilterPath</td>
<td></td>
</tr>
<tr>
<td>• inclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• exclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• exclusionFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionFilePathPatterns</td>
<td></td>
</tr>
<tr>
<td>• exclusionFilePathPatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionOneNoteSectionNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• exclusionOneNoteSectionNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionOneNotePageNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• exclusionOneNotePageNamePatterns</td>
<td></td>
</tr>
<tr>
<td>isUserNameOnS3</td>
<td>true to provide a list of user names in a file stored in an Amazon S3.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify 0NE DRIVE V2 as your data source type.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enableIdentityCrawler</td>
<td>true to activate identity crawler. Identity crawler is activated by default. Crawling identity information on users and groups with access to specific documents is useful for user context filtering. Search results are filtered based on the user or their group access to documents. See <a href="#">Identity crawler</a> for more information.</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index</td>
</tr>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>

### ACL crawling

When you connect an Microsoft OneDrive data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Microsoft OneDrive.
instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

A Microsoft OneDrive data source returns section and page information from OneDrive access control list (ACL) entities. Amazon Q uses the OneDrive tenant domain to connect to the OneDrive instance and can filter based on section name, page type, file name, file type and file contents.

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.

For more information, see:

- Authorization
- Identity crawler
- Understanding User Store

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- (Optional) If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowsAmazonQToGetSecret",
         "Effect": "Allow",
         "Resource": "arn:aws:lambda:*:*"  
      }
   ]
}```
"Action": [  "secretsmanager:GetSecretValue"
],  "Resource": [  "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
],  "Sid": "AllowsAmazonQToDecryptSecret",  "Effect": "Allow",  "Action": [  "kms:Decrypt"
],  "Resource": [  "arn:aws:kms:{{region}}:{{account_id}}:key/[[key_id]]"
],  "Condition": {  "StringLike": {  "kms:ViaService": [  "secretsmanager.*.amazonaws.com"

  ]
  ]
  }
],  "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
],
"Resource": [  
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/
  index/{{index_id}}/"],
"arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/
  index/{{index_id}}/data-source/*"],

{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "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]]"
  ]},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*",
  "Condition": {
    "StringLike": {
      "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "AMAZON_Q"
      ]
    }
  }
},
{
  "Sid": "AllowsAmazonQToCreateTags",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ]}
To allow Amazon Q to assume a role, you must also use the following trust policy:

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


```json
{
   "Sid": "AllowsAmazonQServicePrincipal",
   "Effect": "Allow",
   "Principal": {
      "Service": "qbusiness.amazonaws.com"
   },
   "Action": "sts:AssumeRole",
   "Condition": {
      "StringEquals": {
         "aws:SourceAccount": "{{source_account}}"
      },
      "ArnEquals": {
         "aws:SourceArn": "arn:aws:qbusiness:{{region}}:
          {{source_account}}:application/{{application_id}}"
      }
   }
}
```

For more information on Amazon Q data source connector IAM roles, see [IAM roles for Amazon Q data source connectors](#).

**Connecting Microsoft SharePoint (Cloud) to Amazon Q**

Microsoft SharePoint is a collaborative website building service that lets you customize web content and create web pages, web sites, document libraries, and lists. You can connect SharePoint Online instance to Amazon Q—using either the AWS Management Console or the [CreateDataSource](#) API—and create an Amazon Q web experience.

**Learn more**

- For an overview of the Amazon Q web experience creation process, see [Configuring an application](#).
- For an overview of connector features, see [Data source connector concepts](#).
- For information about connector configuration best practices, see [Connector configuration best practices](#).

**Topics**

Microsoft SharePoint (Cloud)
Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Microsoft SharePoint Online, make sure you have:

- Copied your Microsoft SharePoint Cloud instance URLs. The format for the host URL you enter is `https://yourdomain.sharepoint.com/sites/mysite`. Your URL must start with `https` and contain `sharepoint.com`.
- Copied the domain name of your Microsoft SharePoint Cloud instance URL.
- Noted your Basic authentication credentials containing the username and password that you use to connect to Microsoft SharePoint Cloud Online.
- **If using authentication type other than Basic authentication:** Copied the tenant ID of your Microsoft SharePoint Cloud instance. For details on how to find your tenant ID, see [Find your Microsoft 365 tenant ID](#) on the Microsoft website.
- **For OAuth 2.0 authentication:** Noted your Basic authentication credentials containing the username and password that you use to connect to Microsoft SharePoint Cloud Online and the Client ID and Client secret generated after registering Microsoft SharePoint Cloud with Azure AD.
- **If you're not using ACL**, added the following permissions:

<table>
<thead>
<tr>
<th>Microsoft Graph</th>
<th>SharePoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notes.Read.All (Application) – Read all OneNote notebooks</td>
<td>AllSites.Read (Delegated) – Read items in all site collections</td>
</tr>
<tr>
<td>Sites.Read.All (Application) – Read items in all site collections</td>
<td></td>
</tr>
</tbody>
</table>
### Note

Note.Read.All and Sites.Read.All are required only if you want to crawl OneNote Documents.

- **If you're using ACL**, added the following permissions:

<table>
<thead>
<tr>
<th>Microsoft Graph</th>
<th>SharePoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupMember.Read.All (Application) – Read all group memberships</td>
<td>AllSites.Read (Delegated) – Read items in all site collections</td>
</tr>
<tr>
<td>Notes.Read.All (Application) – Read all OneNote notebooks</td>
<td></td>
</tr>
<tr>
<td>Sites.FullControl.All (Delegated) – Have full control of all site collections</td>
<td></td>
</tr>
<tr>
<td>Sites.Read.All (Application) – Read items in all site collections</td>
<td></td>
</tr>
<tr>
<td>User.Read.All (Application) – Read all users' full profiles</td>
<td></td>
</tr>
</tbody>
</table>

### Note

GroupMember.Read.All and User.Read.All are required only if **Identity crawler** is activated.

- **For Azure AD App-Only authentication**: Noted the X.509 certificate, private key, and the Client ID you generated after registering Microsoft SharePoint Cloud with Azure AD.

- **If you're not using ACL**, added the following permissions:

<table>
<thead>
<tr>
<th>SharePoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllSites.Read (Delegated) – Read items in all site collections</td>
</tr>
</tbody>
</table>
- **SharePoint**
  - Sites.Manage.All (Application) – Read and write items and lists in all site collections

- **If you're using ACL**, added the following permissions:
  - Sites.FullControl.All (Application) – Have full control of all site collections

- **For SharePoint App-Only authentication**: Noted your Microsoft SharePoint Cloud 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 Microsoft SharePoint Cloud app with Azure AD.

- **(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 Microsoft SharePoint Cloud **App-Only authentication**.

In your AWS account, make sure you have:
• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your SharePoint Online authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the SharePoint page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In **Source**, enter the following information:
   
   a. In **Source**, for **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://yourcompany.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.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.
4. For Authentication, choose between Basic, OAuth 2.0, Azure AD App-Only authentication, and SharePoint App-Only authentication based on your use case.

a. If using Basic Authentication, enter the following information:

- For AWS Secrets Manager secret – Choose an existing secret or create a Secrets Manager secret to store your SharePoint authentication credentials. If you choose to create a secret, an AWS Secrets Manager secret window opens. Enter the following information in the window:
  
  - Secret name – A name for your secret.
  - Username – User name for your SharePoint account.
  - Password – Password for your SharePoint account.

b. If using OAuth 2.0 authentication, enter the following information:

- Tenant ID – Tenant ID of your SharePoint account.
- For AWS Secrets Manager secret – Choose an existing secret or create a Secrets Manager secret to store your SharePoint authentication credentials. If you choose to create a secret, an AWS Secrets Manager secret window opens. Enter the following information in the window:
  
  - Secret name – A name for your secret.
  - Username – Username for your SharePoint account.
  - Password – Password for your SharePoint account.
  - Client ID – The Azure AD client ID generated when you register SharePoint in Azure AD.
  - Client secret – The Azure AD client secret generated when you register SharePoint in Azure AD.

c. If using Azure AD App-Only authentication, enter the following information:

- Tenant ID – Tenant ID of your SharePoint account.

---

**Note**

Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).
• **Azure AD self-signed X.509 certificate** – Certificate to authenticate the connector for Azure AD.

• For **AWS Secrets Manager secret** – Choose an existing secret or create a Secrets Manager secret to store your SharePoint authentication credentials. If you choose to create a secret, an AWS Secrets Manager secret window opens. Enter the following information in the window:
  
  • **Secret name** – A name for your secret.
  
  • **Client ID** – The Azure AD client ID generated when you register SharePoint in Azure AD.
  
  • **Private key** – A private key to authenticate the connector for Azure AD.

d. If using **SharePoint App-Only authentication**, enter the following information:

  • **Tenant ID** – Tenant ID of your SharePoint account.

  • For **AWS Secrets Manager secret** — Choose an existing secret or create a Secrets Manager secret to store your SharePoint authentication credentials. If you choose to create a secret, an AWS Secrets Manager secret window opens. Enter the following information in the window:

    • **Secret name** – A name for your secret.

    • **SharePoint client ID** – The SharePoint client ID you generated when you registered App Only at Tenant Level. ClientID format is `ClientID@TenantId`. For example, `ffa956f3-8f89-44e7-b0e4-49670756342c@888d0b57-69f1-4fb8-957f-e1f0bedf82fe`.

    • **SharePoint client secret** – The SharePoint client secret generated when you register for App Only at Tenant Level.

    • **Client ID** – The Azure AD client ID generated when you register SharePoint in Azure AD.

    • **Client secret** – The Azure AD client secret generated when you register SharePoint to Azure AD.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

  a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

For more information, see [VPC](#).

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. Only Local Group Members will be crawled using **Identity crawler**. For more information, see [Identity crawler](#).

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

For more information, see [IAM role](#).

8. In **Sync scope**, choose from the following options:

   a. **Select entities** – Choose the entities that you want to crawl. You can select to crawl All entities or any combination of Files, Attachments, Links, Pages, Events, Comments, and List Data.

   b. In Additional configuration – *optional*, for **Entity regex patterns** – Add regular expression patterns for Links, Pages, and Events to include specific entities instead of syncing all your documents.

   c. In Additional configuration, for **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 patterns.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New or modified content sync** – Sync only new and modified documents.
   - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

   For more details, see [Sync mode](#).

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).
11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See Tags for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see Field mappings.

13. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

   ![Note]

   You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the **CreateDataSource** action to connect a data source to your Amazon Q application.

Then, you use the **configuration** parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**Microsoft SharePoint JSON schema**

The following is the Microsoft 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",
                                "OAuth2_RefreshToken",
                                "Basic"
                            ]
                        }
                    }
                }
            }
        }
    }
}
"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": {
"..."
"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"
              ]
            },
            "dataSourceFieldName": {
              "type": "string"
            },
            "dateFieldFormat": {
              "type": "string",
              "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
            }
          }
        },
        "required": [
          "indexFieldName",
          "indexFieldType",
          "dataSourceFieldName"
        ]
      ]
    }
  }
}
"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": {
            "name": {
              "type": "string",
              "required": true
            },
            "path": {
              "type": "url",
              "required": true
            }
          }
        }
      ]
    }
  }
}
"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"
            }
          }
        }
      ]
    }
  }
}
"indexFieldMap": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "indexFieldName": {
              "type": "string"
            },
            "indexFieldType": {
              "type": "string",
              "enum": [
                "STRING",
                "STRING_LIST",
                "DATE"
              ]
            }
          }
        }
      ]
    }
  }
},
"dataSourceFieldMap": {
  "type": "object",
  "properties": {
    "dateFieldFormat": {
      "type": "string",
      "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
    }
  }
},
"required": [
  "indexFieldName",
  "indexFieldType",
  "dataSourceFieldName"
]
},
"attachment": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "indexFieldName": {
              "type": "string"
            },
            "indexFieldType": {
              "type": "string",
              "enum": [
                "STRING",
                "STRING_LIST",
                "DATE"
              ]
            }
          }
        }
      ]
    }
  }
},
"required": [
  "fieldMappings"
]
"enum": [
  "STRING",
  "STRING_LIST",
  "DATE"
],
"dataSourceFieldName": {
  "type": "string"
},
"dateFieldFormat": {
  "type": "string",
  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
},
"required": [
  "indexFieldName",
  "indexFieldType",
  "dataSourceFieldName"
]
],
"comment": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "indexFieldName": {
              "type": "string"
            },
            "indexFieldType": {
              "type": "string",
              "enum": [
                "STRING",
                "STRING_LIST",
                "DATE"
              ]
            }
          }
        }
      ]
    }
  }
},
"fieldMappings": {
  "type": "array",
  "items": [
    {
      "type": "object",
      "properties": {
        "indexFieldName": {
          "type": "string"
        },
        "indexFieldType": {
          "type": "string",
          "enum": [
            "STRING",
            "STRING_LIST",
            "DATE"
          ]
        }
      }
    }
  ]
}
"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"
      }
    }
  }
}
}
"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"
},
"aclConfiguration": {
  "Microsoft SharePoint (Cloud)"
}
}
"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,
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>tenantId</td>
<td>The tenant id of your SharePoint account.</td>
</tr>
<tr>
<td>domain</td>
<td>The domain of your SharePoint account.</td>
</tr>
<tr>
<td>siteUrls</td>
<td>The host URLs of your SharePoint account.</td>
</tr>
<tr>
<td>repositoryAdditionalProperties</td>
<td>Additional properties to connect with your repository endpoint.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>s3bucketName</td>
<td>The name of the Amazon S3 bucket that stores your Azure AD self-signed X.509 certificate.</td>
</tr>
<tr>
<td>s3certificateName</td>
<td>The name of the SSL certificate stored in your Amazon S3 bucket.</td>
</tr>
<tr>
<td>authType</td>
<td>The type of authentication you are using: OAuth2, OAuth2Certificate, OAuth2App, or Basic.</td>
</tr>
<tr>
<td>version</td>
<td>The SharePoint version you are using: Online.</td>
</tr>
<tr>
<td>onPremVersion</td>
<td>Not required if you are using SharePoint Online.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• event</td>
<td>A list of objects that map the attributes or field names of your Microsoft SharePoint pages and assets to Amazon Q index field names.</td>
</tr>
<tr>
<td>• page</td>
<td></td>
</tr>
<tr>
<td>• file</td>
<td></td>
</tr>
<tr>
<td>• link</td>
<td></td>
</tr>
<tr>
<td>• attachment</td>
<td></td>
</tr>
<tr>
<td>• comment</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>eventTitleFilterRegEx</td>
<td>A list of regular expression patterns to include/exclude specific files in your SharePoint 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 precedence, and the file isn't included in the index.</td>
</tr>
<tr>
<td>pageTitleFilterRegEx</td>
<td></td>
</tr>
<tr>
<td>linkTitleFilterRegEx</td>
<td></td>
</tr>
<tr>
<td>inclusionFilePath</td>
<td></td>
</tr>
<tr>
<td>exclusionFilePath</td>
<td></td>
</tr>
<tr>
<td>inclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionOneNoteSectionNamePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionOneNoteSectionNamePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionOneNotePageNamePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionOneNotePageNamePatterns</td>
<td></td>
</tr>
<tr>
<td>aclConfiguration</td>
<td></td>
</tr>
<tr>
<td>emailDomain</td>
<td></td>
</tr>
<tr>
<td>proxyHost</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>proxyPort</td>
<td></td>
</tr>
<tr>
<td>• crawlFiles</td>
<td>Input TRUE to index.</td>
</tr>
<tr>
<td>• crawlPages</td>
<td></td>
</tr>
<tr>
<td>• crawlEvents</td>
<td></td>
</tr>
<tr>
<td>• crawlComments</td>
<td></td>
</tr>
<tr>
<td>• crawlLinks</td>
<td></td>
</tr>
<tr>
<td>• crawlAttachments</td>
<td></td>
</tr>
<tr>
<td>• crawlListData</td>
<td></td>
</tr>
<tr>
<td>• crawlAcl</td>
<td></td>
</tr>
<tr>
<td>• iscrawlLocalGroupMapping</td>
<td></td>
</tr>
<tr>
<td>• iscrawlAdGroupMapping</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>Specify SHAREPOINTV2 as your data source type</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>true to activate identity crawler. Identity crawler is activated by default. Crawling identity information on users and groups with access to specific documents is useful for user context filtering. Search results are filtered based on the user or their group access to documents. See Identity crawler for more information.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index</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 SharePoint. If you use basic authentication provide the username and password. If you use OAuth 2.0 authentication, provide the username, password, client ID, and client secret.</td>
</tr>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>

**ACL crawling**

When you connect an Microsoft SharePoint data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Microsoft SharePoint.
instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

To filter using a user name, use the **User principal name** from your Azure portal. For example, johnstiles@kendra.onmicrosoft.com.

When you use a SharePoint group for user context filtering, 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 "site owners", the group ID is the following: "430a6b90503eef95c89295c8999c7981|site owners".

For more information, see:
- Authorization
- Identity crawler
- Understanding User Store

**IAM roles**

To connect SharePoint Online to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the **User Store** operations to ingest access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your SharePoint Online instance.
- **(Optional)** If you use Azure App-Only authentication, permission to access the certificate stored in your Amazon S3 bucket.
• **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowsAmazonQToGetS3Objects",
        "Action": [
            "s3:GetObject"
        ],
        "Resource": [
            "arn:aws:s3:::{{input_bucket_name}}/*"
        ],
        "Effect": "Allow",
        "Condition": {
            "StringEquals": {
                "aws:ResourceAccount": "{{account_id}}"
            }
        }
    },
    {
        "Sid": "AllowsAmazonQToGetSecret",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:GetSecretValue"
        ],
        "Resource": [
            "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:{{secret_id}}"
        ]
    },
    {
        "Sid": "AllowsAmazonQToDecryptSecret",
        "Effect": "Allow",
        "Action": [
            "kms:Decrypt"
        ],
        "Resource": [
            "arn:aws:kms:{{region}}:{{account_id}}:key:{{key_id}}"
        ],
        "Condition": {
            "StringLike": {
                "kms:ViaService": [
                    "secretsmanager.*.amazonaws.com"
                ]
            }
        }
    }
}
```
"Sid": "AllowsAmazonQToIngestDocuments",
"Effect": "Allow",
"Action": [
  "qbusiness:BatchPutDocument",
  "qbusiness:BatchDeleteDocument"
],
"Resource": "arn:aws:qbusiness:{region}:{source_account}:application/{application_id}/index/{index_id}"
},
{
"Sid": "AllowsAmazonQToIngestPrincipalMapping",
"Effect": "Allow",
"Action": [
  "qbusiness:PutGroup",
  "qbusiness:CreateUser",
  "qbusiness:DeleteGroup",
  "qbusiness:UpdateUser",
  "qbusiness:ListGroups"
],
"Resource": [
  "arn:aws:qbusiness:{region}:{account_id}:application/{application_id}",
  "arn:aws:qbusiness:{region}:{account_id}:application/{application_id}/index/{index_id}"
]
},
{
"Sid": "AllowsAmazonQToCreateAndDeleteNI",
"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]"
]
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
            "Effect": "Allow",
            "Principal": {
                "Service": "qbusiness.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "{{source_account}}"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
                }
            }
        }
    ]
}
```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting Microsoft SharePoint (Server) to Amazon Q

Microsoft SharePoint is a collaborative website building service that lets you customize web content and create web pages, web sites, document libraries, and lists. You can connect SharePoint Server instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.


Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites:

In Microsoft SharePoint Server, make sure you have:
• Copied your Microsoft SharePoint Server instance URLs and the domain name of your Microsoft SharePoint Server URLs. The format for the host URL you enter is https://yourcompany/sites/mysite. Your URL must start with https.

• If using **SharePoint App-Only authentication** for access control:
  
  • Copied the SharePoint client ID generated when you registered App Only at Site Level. ClientID 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.
  
  • Noted the Tenant ID of your SharePoint Server account.

  **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.

  ![](image)

  *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 **IAM role** for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
• Stored your SharePoint Server authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the SharePoint page, enter the following information:

1. **Name** – Name your data source for easy tracking.

   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Source**, enter the following information:

   a. **Hosting Method** – Choose SharePoint Server.


   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.

   e. **SSL certificate location** – Enter the Amazon S3 path to your SSL certificate file.

3. **Web proxy – optional** – 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.

4. **Authorization** – You can choose to use an access control list (ACL) for controlling search results based on your end user's document access level in your SharePoint data source. Authorization using ACL is activated by default. When ACL is deactivated, no ACL information
is crawled and no access control or context filtering is available. For SharePoint Server, you can choose from the following ACL options:

a. **Email ID with Domain from IDP** – Access control is based on email IDs that are extracted from email domains fetched from the underlying identity provider (IdP). You provide the IdP connection details in your Secrets Manager secret during **Authentication**.

b. **Email ID with Custom Domain** – Access control is based on email IDs. Provide the email domain value. For example, "amazon.com". The email domain is used to construct the email ID for access control. You must enter your email domain using **Add Email Domain**.

See **Authorization** for more details.

5. For **Authentication**, choose between **SharePoint App-Only authentication**, **NTLM authentication**, and **Kerberos authentication**, based on your use case.

a. Enter the following information for both **NTLM authentication** and **Kerberos authentication**:

For **AWS Secrets Manager secret** – Choose an existing secret or create a Secrets Manager secret to store your SharePoint authentication credentials. If you choose to create a secret, an AWS Secrets Manager secret window opens. Enter the following information in the window:

- **Secret name** – A name for your secret.
- **Username** – Username for your SharePoint account.
- **Password** – Password for your SharePoint account.

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 username.
- **LDAP Password** – Your LDAP password.

b. Enter the following information for **SharePoint App-Only authentication**:
For **AWS Secrets Manager secret** – Choose an existing secret or create a Secrets Manager secret to store your SharePoint authentication credentials. If you choose to create a secret, an AWS Secrets Manager secret window opens. Enter the following information in the window:

- **Secret name** – A name for your secret.
- **Client ID** – The SharePoint client ID that you generated when you registered App Only at Site Level. The ClientID format is ClientID@TenantId. For example, `ffa956f3-8f89-44e7-b0e4-49670756342c@888d0b57-69f1-4fb8-957f-e1f0bedf82fe`.
- **SharePoint client secret** – The SharePoint client secret generated when your register for 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.

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.

6. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).
7. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. Only **Local Group Members** will be crawled by **Identity crawler**. For more information, see [Identity crawler](#).

8. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

For more information, see [IAM role](#).

9. In **Sync scope**, choose from the following options:
   a. **Select entities** – Choose the entities that you want to crawl. You can select to crawl **All** entities or any combination of **Files**, **Attachments**, **Links**, **Pages**, **Events** and **List Data**.
   b. In **Additional configuration** – **optional**, for **Entity regex patterns** – Add regular expression patterns for **Links**, **Pages**, and **Events** to include specific entities instead of syncing all your documents.
   c. **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 patterns.

10. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.
    - **Full sync** – Sync all content regardless of the previous sync status.
    - **New or modified content sync** – Sync only new and modified documents.
    - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

    For more details, see [Sync mode](#).

11. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

12. **Tags** – **optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

13. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:
a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

For more information, see Field mappings.

14. To finish connecting your data source to Amazon Q, select **Add data source**.

You are taken to the **Data source details**, where you can view your data source configuration details.

15. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

**Note**

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the **CreateDataSource** action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**Microsoft SharePoint JSON schema**

The following is the Microsoft SharePoint JSON schema:

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "connectionConfiguration": {
            "properties": {
                "connectionConfiguration": {
```
"type": "object",
"properties": {
"repositoryEndpointMetadata": {
"type": "object",
"properties": {
"tenantId": {
"type": "string",
"pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$",
"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",
"OAuth2_RefreshToken",
"Basic",
"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": {
                        "field": {
                           "type": "string"
                        }
                     }
                  }
               ]
            }
         }
      }
   }
}
"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"
            }
          }
        }]
      }
    }
  }
}
"indexFieldName": {
  "type": "string"
},
"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"
            }
          }
        }
      ]
    }
  }
}
"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",
"NUMBER",
"DATE",
"LONG"
]
},
"required": [
"fieldMappings"
]
"link": {
"type": "object",
"properties": {
"fieldMappings": {
"type": "array",
"items": [
{
"type": "object",
"properties": {
"indexFieldName": {
"type": "string"
},
"indexFieldType": {
"type": "string",
"enum": [
"STRING",
"STRING_LIST",
"NUMBER",
"DATE",
"LONG"
]
}
"DATE"
],
"dataSourceFieldName": {
  "type": "string"
},
"dateFieldFormat": {
  "type": "string",
  "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
}
],
"required": [
  "indexFieldName",
  "indexFieldType",
  "dataSourceFieldName"
]
],
"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"
]}
],
"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": [
  "indexFieldId",
  "indexFieldId",
  "dataSourceFieldId"
],
"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"
      }
    }
  }
}
"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"
},
"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": [

The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>tenantId</td>
<td>The tenant id of your SharePoint account.</td>
</tr>
<tr>
<td>domain</td>
<td>The domain of your SharePoint account.</td>
</tr>
<tr>
<td>siteUrls</td>
<td>The host URLs of your SharePoint account.</td>
</tr>
<tr>
<td>repositoryAdditionalProperties</td>
<td>Additional properties to connect with your repository endpoint.</td>
</tr>
<tr>
<td>s3bucketName</td>
<td>The name of the Amazon S3 bucket that stores your Azure AD self-signed X.509 certificate.</td>
</tr>
<tr>
<td>s3certificateName</td>
<td>The name of the SSL certificate stored in your Amazon S3 bucket.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>authType</td>
<td>The type of authentication you are using: OAuth2, OAuth2Certificate, OAuth2App, or Basic.</td>
</tr>
<tr>
<td>version</td>
<td>The SharePoint version you are using: Online.</td>
</tr>
<tr>
<td>onPremVersion</td>
<td>Not required if you are using SharePoint Online.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• event</td>
<td>A list of objects that map the attributes or field names of your Microsoft SharePoint pages and assets to Amazon Q index field names.</td>
</tr>
<tr>
<td>• page</td>
<td></td>
</tr>
<tr>
<td>• file</td>
<td></td>
</tr>
<tr>
<td>• link</td>
<td></td>
</tr>
<tr>
<td>• attachment</td>
<td></td>
</tr>
<tr>
<td>• comment</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>eventTitleFilterRegEx</td>
<td>A list of regular expression patterns to include/exclude specific files in your SharePoint 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 precedence, and the file isn't included in the index.</td>
</tr>
<tr>
<td>pageTitleFilterRegEx</td>
<td></td>
</tr>
<tr>
<td>linkTitleFilterRegEx</td>
<td></td>
</tr>
<tr>
<td>inclusionFilePath</td>
<td></td>
</tr>
<tr>
<td>exclusionFilePath</td>
<td></td>
</tr>
<tr>
<td>inclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionOneNoteSectionNamePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionOneNoteSectionNamePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionOneNotePageNamePatterns</td>
<td></td>
</tr>
<tr>
<td>exclusionOneNotePageNamePatterns</td>
<td></td>
</tr>
<tr>
<td>aclConfiguration</td>
<td></td>
</tr>
<tr>
<td>emailDomain</td>
<td></td>
</tr>
<tr>
<td>proxyHost</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• proxyPort</td>
<td></td>
</tr>
<tr>
<td>• crawlFiles</td>
<td></td>
</tr>
<tr>
<td>• crawlPages</td>
<td></td>
</tr>
<tr>
<td>• crawlEvents</td>
<td></td>
</tr>
<tr>
<td>• crawlComments</td>
<td></td>
</tr>
<tr>
<td>• crawlLinks</td>
<td></td>
</tr>
<tr>
<td>• crawlAttachments</td>
<td></td>
</tr>
<tr>
<td>• crawlListData</td>
<td></td>
</tr>
<tr>
<td>• crawlAcl</td>
<td></td>
</tr>
<tr>
<td>• isCrawlLocalGroupMapping</td>
<td></td>
</tr>
<tr>
<td>• isCrawlAdGroupMapping</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>Specify SHAREPOINTV2 as your data source type</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>true to activate identity crawler. Identity crawler is activated by default. Crawling identity information on users and groups with access to specific documents is useful for user context filtering. Search results are filtered based on the user or their group access to documents. See <a href="#">Identity crawler</a> for more information.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>syncMode</strong></td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index</td>
</tr>
<tr>
<td><strong>secretARN</strong></td>
<td>The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key-value pairs required to connect to your SharePoint. If you use basic authentication provide the username and password. If you use OAuth 2.0 authentication, provide the username, password, client ID, and client secret.</td>
</tr>
<tr>
<td><strong>version</strong></td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>

### ACL crawling

When you connect an Microsoft SharePoint data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Microsoft SharePoint.
instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

To filter using a user name, use the **User principal name** from your Azure portal. For example, johnstiles@kendra.onmicrosoft.com.

When you use a SharePoint group for user context filtering, 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 "site owners", the group ID is the following: 

   "430a6b90503eef95c89295c8999c7981|site owners"

For more information, see:

- **Authorization**
- **Identity crawler**
- **Understanding User Store**

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.  
- Permission to access the **User Store** API operations to ingest user and group access control information from documents.  
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.  
- Permission to access the SSL certificate stored in your Amazon S3 bucket.  
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowsAmazonQToGetS3Objects",
         "Action": [ "s3:GetObject" ],
         "Resource": [ "arn:aws:s3:::{{input_bucket_name}}/*" ],
         "Effect": "Allow",
         "Condition": {
            "StringEquals": {
               "aws:ResourceAccount": "{{account_id}}"
            }
         }
      },
      {
         "Sid": "AllowsAmazonQToGetSecret",
         "Effect": "Allow",
         "Action": [ "secretsmanager:GetSecretValue" ],
         "Resource": [ "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]" ]
      },
      {
         "Sid": "AllowsAmazonQToDecryptSecret",
         "Effect": "Allow",
         "Action": [ "kms:Decrypt" ],
         "Resource": [ "arn:aws:kms:{{region}}:{{account_id}}:key:[[key_id]]" ],
         "Condition": {
            "StringLike": {
               "kms:ViaService": [ "secretsmanager.*.amazonaws.com"
            ]
         }
      }
   ]
}
"Sid": "AllowsAmazonQToIngestDocuments",
"Effect": "Allow",
"Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
],
"Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},

"Sid": "AllowsAmazonQToIngestPrincipalMapping",
"Effect": "Allow",
"Action": [
    "qbusiness:PutGroup",
    "qbusiness:CreateUser",
    "qbusiness:DeleteGroup",
    "qbusiness:UpdateUser",
    "qbusiness:ListGroups"
],
"Resource": [
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
    "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/*"
]
},

"Sid": "AllowsAmazonQToCreateAndDeleteNI",
"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]]"
]
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*/",
"Condition": {
    "StringLike": {
        "aws:RequestTag/AMAZON_Q":
        "qbusiness_{{account_id}}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
    "aws:TagKeys": [
        "AMAZON_Q"
    ]
    }
},
},
{
    "Sid": "AllowsAmazonQToCreateTags",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*/",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "CreateNetworkInterface"
        }
    }
},
},
{
    "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterfacePermission"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/*/",
    "Condition": {
        "StringLike": {
            "aws:ResourceTag/AMAZON_Q":
            "qbusiness_{{account_id}}_{{application_id}}_*"
        }
    }
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
        }
      }
    }
  ]
}
```
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting Microsoft SQL Server to Amazon Q

Microsoft SQL Server is an relational database management system (RDBMS) developed by Microsoft. You can connect your Microsoft SQL Server instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

The Amazon Q Microsoft SQL Server data source connector supports MS SQL Server 2019.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Microsoft SQL Server, make sure you have:

- Noted your database user name and password.
**Important**

As a best practice, provide Amazon Q with read-only database credentials.

- Copied your database host URL, port, and instance.

**In your AWS account, make sure you have:**

- Created an [IAM role](#) for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

- Stored your Microsoft SQL Server authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

**Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

**Using the console**

On the **Microsoft SQL Server** page, enter the following information:

1. **Name** – Name your data source for easy tracking.

   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In **Source**, enter the following information:

   a. **Host** – Enter the database host name.

   b. **Port** – Enter the database port.

   c. **Instance** – Enter the database instance.

   d. **Enable SSL certificate location** – Choose to enter the Amazon S3 path to your SSL certificate file.
3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.

   > **Note**
   > Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).

4. In **Authentication** – Enter the following information for your AWS Secrets Manager secret.
   
   a. **Secret name** – A name for your secret.
   
   b. For **Database user name** and **Password** – Enter the authentication credential values you copied from your database.
   
   c. Choose **Save**.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:
   
   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

7. In **Sync scope**, enter the following information:
   
   - **SQL query** – Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Q 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 names of the document body column within your database table.

8. In **Additional configuration – optional** – Configure the following settings:

• **Change-detecting columns** – Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q 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 Q 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.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

• **Full sync** – Sync all content regardless of the previous sync status.

• **New or modified content sync** – Sync only new and modified documents.

• **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

For more details, see **Sync mode**.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see **Sync run schedule**.

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See **Tags** for more details.
12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

   ![Note]

   You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**JSON schema**

The following is the JSON schema:

```json
{
   "$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"
"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"
},
"serverlessAurora": {
    "type": "string",
    "enum": ["true", "false"]
},
"required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
"type": {
    "type": "string",
    "pattern": "JDBC"
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>Required configuration information for connecting your data source.</td>
</tr>
<tr>
<td></td>
<td>• dbType—The type of Java database you are using, whether mysql, db2, postgresql, oracle, or sqlserver.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>dbHost</strong></td>
<td>The database host name.</td>
</tr>
<tr>
<td><strong>dbPort</strong></td>
<td>The database port.</td>
</tr>
<tr>
<td><strong>dbInstance</strong></td>
<td>The database instance.</td>
</tr>
<tr>
<td><strong>repositoryConfigurations</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>document</strong></td>
<td>A list of objects that map the attributes or field names of your database content to Amazon Q index field names. For more information, see <a href="#">Mapping data source fields</a>.</td>
</tr>
<tr>
<td><strong>additionalProperties</strong></td>
<td>Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.</td>
</tr>
<tr>
<td><strong>primaryKey</strong></td>
<td>Provide the primary key for the database table. This identifies a table within your database.</td>
</tr>
<tr>
<td><strong>titleColumn</strong></td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td><strong>bodyColumn</strong></td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td><strong>sqlQuery</strong></td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>timestampColumn</td>
<td>Enter the name of the column which contains time stamps. Amazon Q uses time stamp information to detect changes in your content and sync only changed content.</td>
</tr>
<tr>
<td>timestampFormat</td>
<td>Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.</td>
</tr>
<tr>
<td>timezone</td>
<td>Enter the name of the column which contains time zones for the content to be crawled.</td>
</tr>
<tr>
<td>changeDetectingColumns</td>
<td>Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q will re-index content when there is a change in any of these columns</td>
</tr>
<tr>
<td>allowedUsersColumns</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>allowedGroupsColumn</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>sourceURIColumn</td>
<td>Enter the name of the column which contains Source URLs to be indexed.</td>
</tr>
<tr>
<td>isSslEnabled</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify JDBC as your data source type.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| syncMode      | Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose:  
|               | • FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index  
|               | • FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index  
|               | • CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index. |
| 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. |

**ACL crawling**

When you connect a database data source to Amazon Q, Amazon Q crawls user and group information from a column in the source table. You specify this column in the console or using the configuration parameter as part of the CreateDataSource operation.
If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- Authorization
- Identity crawler
- Understanding User Store

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- Permission to access the SSL certificate stored in your Amazon S3 bucket.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowsAmazonQToGetS3Objects",
        "Action": ["s3:GetObject"
    ]},
```
"Resource": [
    "arn:aws:s3:::{input_bucket_name}/*"
],
"Effect": "Allow",
"Condition": {
    "StringEquals": {
        "aws:ResourceAccount": "{{account_id}}"
    }
}
},
{
    "Sid": "AllowsAmazonQToGetSecret",
    "Effect": "Allow",
    "Action": [
        "secretsmanager:GetSecretValue"
    ],
    "Resource": [
        "arn:aws:secretsmanager:{region}:{account_id}:secret:[[secret_id]]"
    ]
},
{
    "Sid": "AllowsAmazonQToDecryptSecret",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:{region}:{account_id}:key:[[key_id]]"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": ["secretsmanager.*.amazonaws.com"
        ]
    }
}
},
{
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ]
}
"Resource": "arn:aws:qbusiness:{region}::{source_account}:application/
{{application_id}}/index/{{index_id}}",

{{application_id}}",
 "arn:aws:qbusiness:{region}::{account_id}:application/
{{application_id}}/index/{{index_id}}",
 "arn:aws:qbusiness:{region}::{account_id}:application/
{{application_id}}/index/{{index_id}}/data-source/*" ] }

 "arn:aws:ec2:{region}::{account_id}:security-group/[[security_group]]" ] }

{ "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag", "Effect": "Allow", "Action": [ "ec2:CreateNetworkInterface", "ec2:DeleteNetworkInterface" ], "Resource": "arn:aws:ec2:{region}::{account_id}:network-interface/*", "Condition": { } }
"StringLike": {
  "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_**",
  "ForAllValues:StringEquals": {
    "aws:TagKeys": [
      "AMAZON_Q"
    ]
  }
},
{
  "Sid": "AllowsAmazonQToCreateTags",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
    }
  }
},
{
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_**"
    }
  }
},
{
  "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeAvailabilityZones"
  ]
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
         "Effect": "Allow",
         "Principal": {
            "Service": "qbusiness.amazonaws.com"
         },
         "Action": "sts:AssumeRole",
         "Condition": {
            "StringEquals": {
               "aws:SourceAccount": "{{source_account}}"
            },
            "ArnLike": {
               "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
            }
         }
      }
   ]
}
```

For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.
Connecting Microsoft Teams to Amazon Q

Microsoft Teams is an enterprise collaboration tool for messaging, meetings, and file sharing. You can connect Microsoft Teams instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

• For an overview of the Amazon Q web experience creation process, see Configuring an application.
• For an overview of connector features, see Data source connector concepts.
• For information about connector configuration best practices, see Connector configuration best practices.

Topics

• Prerequisites
• Using the console
• Using the API
• ACL crawling
• IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Microsoft Teams, make sure you have:

• Created a Microsoft Teams account in Office 365.
• Copied your Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal. You need this URL to allow Amazon Q to connect with your Microsoft Teams data source.
• Configured an OAuth 2.0 credential token containing a client ID and client secret. For more information, see Azure documentation on managing access tokens for Teams on the Microsoft website.
• 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 want to crawl. The following table shows permissions by corresponding entity.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Required permissions for data sync</th>
<th>Required permissions for identity sync</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel Post</td>
<td>• ChannelMessage.Read.All</td>
<td>TeamMember.Read.All</td>
</tr>
<tr>
<td></td>
<td>• Group.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• User.Read</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• User.Read.All</td>
<td></td>
</tr>
<tr>
<td>Channel Attachment</td>
<td>• ChannelMessage.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Group.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• User.Read</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• User.Read.All</td>
<td></td>
</tr>
<tr>
<td>Channel Wiki</td>
<td>• Group.Read.All</td>
<td>TeamMember.Read.All</td>
</tr>
<tr>
<td></td>
<td>• User.Read</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• User.Read.All</td>
<td></td>
</tr>
<tr>
<td>Chat Message</td>
<td>• Chat.Read.All</td>
<td>TeamMember.Read.All</td>
</tr>
<tr>
<td></td>
<td>• ChatMessage.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ChatMember.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• User.Read</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• User.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Group.Read.All</td>
<td></td>
</tr>
<tr>
<td>Meeting Chat</td>
<td>• Chat.Read.All</td>
<td>TeamMember.Read.All</td>
</tr>
<tr>
<td></td>
<td>• ChatMessage.Read</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ChatMember.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• User.Read</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• User.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Group.Read.All</td>
<td></td>
</tr>
<tr>
<td>Entity</td>
<td>Required permissions for data sync</td>
<td>Required permissions for identity sync</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Chat Attachment</td>
<td>Chat.Read.All</td>
<td>TeamMember.Read.All</td>
</tr>
<tr>
<td></td>
<td>ChatMessage.Read</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ChatMember.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>User.Read</td>
<td></td>
</tr>
<tr>
<td></td>
<td>User.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group.Read.All</td>
<td></td>
</tr>
<tr>
<td>Meeting File</td>
<td>Chat.Read.All</td>
<td>TeamMember.Read.All</td>
</tr>
<tr>
<td></td>
<td>ChatMessage.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ChatMember.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>User.Read</td>
<td></td>
</tr>
<tr>
<td></td>
<td>User.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Files.Read.All</td>
<td></td>
</tr>
<tr>
<td>Calendar Meeting</td>
<td>Chat.Read.All</td>
<td>TeamMember.Read.All</td>
</tr>
<tr>
<td></td>
<td>ChatMessage.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ChatMember.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>User.Read</td>
<td></td>
</tr>
<tr>
<td></td>
<td>User.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Files.Read.All</td>
<td></td>
</tr>
<tr>
<td>Meeting Notes</td>
<td>User.Read</td>
<td>TeamMember.Read.All</td>
</tr>
<tr>
<td></td>
<td>User.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group.Read.All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Files.Read.All</td>
<td></td>
</tr>
</tbody>
</table>

- Generated Microsoft Teams OAuth 2.0 credentials containing a client id, client secret, username, and password. You need these credentials to authenticate Amazon Q to access Microsoft Teams.
In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your Microsoft Teams authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Teams page, enter the following information:

1. **Name** – Name your data source for easy tracking.

   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **In Source,** enter the following information:

   - **Tenant ID** – Enter your tenant id. Your Microsoft tenant ID is a globally unique identifier that's necessary to configure each connector instance. Your tenant ID is different from your organization name or domain and can be found in the properties section of your Microsoft account dashboard.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.

Note

Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).
4. **Authentication** – Choose between **New** and **Existing**.

   - If you choose **Existing**, select an existing secret for **Select secret**.

     If you choose **New**, enter the following information in the **New AWS Secrets Manager secret** section:

     i. **Secret name** – A name for your secret.

     ii. For **Client ID, Client secret** – Enter the authentication credential values that you generated from your Teams account.

5. **Payment model** – You can choose a licensing and payment model for your 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 don't require security compliance.

6. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

7. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

8. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

9. **Sync scope** – Select the content you want to sync.

10. In **Additional configuration – optional**, choose from the following options:

    - **Calendar crawling** – Enter the date range for which the connector will crawl your calendar content.

    - **User email** – Enter the user emails you wish to include in your application.
• **Team names** – Add patterns to include or exclude teams found in Microsoft Teams from your application.

• **Channel names** – Add patterns to include or exclude channels found in Microsoft Teams from your application.

• **Attachment regex patterns** – Add regular expression patterns to include or exclude certain attachment for all supported entities. You can add up to 100 patterns.

11. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   • **Full sync** – Sync all content regardless of the previous sync status.
   
   • **New or modified content sync** – Sync only new and modified documents.
   
   • **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

   For more details, see [Sync mode](#).

12. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

13. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

14. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can’t edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

15. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.
16. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

**Note**

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the `configuration` parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**Microsoft Teams JSON schema**

The following is the Microsoft Teams JSON schema:

```json
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "connectionConfiguration": {
      "type": "object",
      "properties": {
        "repositoryEndpointMetadata": {
          "type": "object",
          "properties": {
            "tenantId": {
              "type": "string",
              "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$",
              "minLength": 36,
              "maxLength": 36
            }
          }
        }
      }
    }
  }
}
```

"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'"
                }
              }
            },
            {
              "type": "object",
              "properties": {
                "indexFieldName": {
                  "type": "string"
                },
                "indexFieldType": {
                  "type": "string",
                  "enum": ["STRING", "STRING_LIST", "DATE"]
                },
                "dataSourceFieldName": {
                  "type": "string"
                }
              }
            }
          ]
        }
      },
      "required": [
        "indexFieldName",
        "indexFieldType",
        "dataSourceFieldName"
      ]
    }
  }
}


```json
},
]
},
"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": ["channelWiki"]
}
"type": "object",
"properties": {
  "fieldMappings": {
    "type": "array",
    "items": [
      {
        "type": "object",
        "properties": {
          "indexFieldName": {
            "type": "string"
          },
          "indexFieldFormat": {
            "type": "string",
            "enum": [
              "STRING",
              "DATE",
              "LONG"
            ],
          },
        },
        "required": [
          "indexFieldName",
          "indexFieldFormat"
        ]
      }
    ],
    "required": ["fieldMappings"]
  }
},
"meetingChat": {
  "type": "object",
  "properties": {
    "fieldMappings": {
      "type": "object",
      "properties": {
        "indexFieldName": {
          "type": "string"
        },
        "dateFieldFormat": {
          "type": "string",
          "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
        }
      },
      "required": ["indexFieldName", "dateFieldFormat"]
    }
  }
}}
"type": "array",
"items": [
  {
    "type": "object",
    "properties": {
      "indexFieldName": {
        "type": "string"
      },
      "indexFieldType": {
        "type": "string",
        "enum": [
          "STRING",
          "STRING_LIST",
          "DATE"
        ]
      },
      "dataSourceFieldName": {
        "type": "string"
      },
      "dateFormat": {
        "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",
                "STRING_LIST",
                "DATE"
              ]
            },
            "dataSourceFieldName": {
              "type": "string"
            },
            "dateFormat": {
              "type": "string",
              "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
            }
          },
          "required": [
            "indexFieldName",
            "indexFieldType",
            "dataSourceFieldName"
          ]
        }
      ]
    }
  }
}
"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"
            }
          }
        }
      ]
    }
  }
}
"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": {
        "isCrawlAcl": {
            "type": "boolean"
        },
        "fieldForUserId": {
            "type": "string"
        },
        "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"
  }
}
"exclusionFileTypePatterns": {
  "type": "array",
  "items": {
    "type": "string"
  }
},
"inclusionUserEmailFilter": {
  "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",
      "Microsoft Teams": 735
    }
  ]
}
"pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}Z$",
]
],
"endCalendarDateTime": {
"anyOf": [
{"type": "string",
"pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}Z$",
}],
{"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
}
}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>tenantId</td>
<td>The Microsoft 365 tenant ID. You can find your tenant ID in the Properties of your Azure Active Directory Portal.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• chatMessage</td>
<td>A list of objects that map the attributes or field names of your Adobe Experience Manager pages and assets to Amazon Q index field names.</td>
</tr>
<tr>
<td>• chatAttachment</td>
<td></td>
</tr>
<tr>
<td>• channelPost</td>
<td></td>
</tr>
<tr>
<td>• channelWiki</td>
<td></td>
</tr>
<tr>
<td>• channelAttachment</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>• meetingChat</td>
<td></td>
</tr>
<tr>
<td>• meetingFile</td>
<td></td>
</tr>
<tr>
<td>• meetingNote</td>
<td></td>
</tr>
<tr>
<td>• calendarMeeting</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>• isCrawlChatMessage</td>
<td>true to index this content in your Microsoft Teams data source.</td>
</tr>
<tr>
<td>• isCrawlChatAttachment</td>
<td></td>
</tr>
<tr>
<td>• isCrawlChannelPost</td>
<td></td>
</tr>
<tr>
<td>• isCrawlChannelAttachment</td>
<td></td>
</tr>
<tr>
<td>• isCrawlChannelWiki</td>
<td></td>
</tr>
<tr>
<td>• isCrawlCalendarMeeting</td>
<td></td>
</tr>
<tr>
<td>• isCrawlMeetingChat</td>
<td></td>
</tr>
<tr>
<td>• isCrawlMeetingFile</td>
<td></td>
</tr>
<tr>
<td>• isCrawlMeetingNote</td>
<td></td>
</tr>
<tr>
<td>paymentModel</td>
<td>Specifies what type of payment model to use with your 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 don't require security compliance.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index</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 Teams. This includes your client ID and client secret.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify MSTEAMS as your data source type.</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>true to activate identity crawler. Crawling identity information on users and groups with access to specific documents is useful for user context filtering. Search results are filtered based on the user or their group access to documents. See Identity crawler for more information.</td>
</tr>
</tbody>
</table>
**Configuration** | **Description**
---|---
version | The version of this template that's currently supported.

**ACL crawling**

When you connect an Microsoft Teams data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Microsoft Teams instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

The group and user IDs are mapped as follows:

- `_tenant_id` – Your Microsoft tenant ID is a globally unique identifier that's necessary to configure each connector instance. Your tenant ID is different from your organization name or domain and can be found in the properties section of your Microsoft account dashboard.

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the [User Store](#) API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- *(Optional)* If you're using Amazon VPC, permission to access your Amazon VPC.
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToGetSecret",
      "Effect": "Allow",
      "Action": ["secretsmanager:GetSecretValue"],
      "Resource": ["arn:aws:secretsmanager:{region}:{account_id}:secret:[[secret_id]]"]
    },
    {
      "Sid": "AllowsAmazonQToDecryptSecret",
      "Effect": "Allow",
      "Action": ["kms:Decrypt"],
      "Resource": ["arn:aws:kms:{region}:{account_id}:key/[key_id]"],
      "Condition": {
        "StringLike": {
          "kms:ViaService": ["secretsmanager.*.amazonaws.com"
        ]
      }
    },
    {
      "Sid": "AllowsAmazonQToIngestDocuments",
      "Effect": "Allow",
      "Resource": "arn:aws:qbusiness:{region}:{source_account}:application/{application_id}/index/{index_id}"}
  ]
}
"Resource": [  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/*"  ]},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "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]]"  ]},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
  "Effect": "Allow",
  "Action": [  "ec2:CreateNetworkInterface",  "ec2:DeleteNetworkInterface"  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {  "StringLike": {  "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"  },  "ForAllValues:StringEquals": {  "aws:TagKeys": [  "AMAZON_Q"  ]  }  }  }
"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
  "ec2:CreateTags"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
  "StringEquals": {
    "ec2:CreateAction": "CreateNetworkInterface"
  }
}
},
{
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    }
  }
},
{
  "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeNetworkInterfaceAttribute",
    "ec2:DescribeVpcs",
    "ec2:DescribeRegions",
    "ec2:DescribeNetworkInterfacePermissions",
    "ec2:DescribeSubnets"
  ],
  "Resource": "**"
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
        }
      }
    }
  ]
}
```

For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

**Connecting Microsoft Yammer to Amazon Q**

Microsoft Yammer is an enterprise collaboration tool for messaging, meetings, and file sharing. You can connect Microsoft Yammer instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

**Learn more**

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
For information about connector configuration best practices, see [Connector configuration best practices](#).

### Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

### Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

**In Microsoft Yammer, make sure you have:**

- Created a Microsoft Yammer administrative account.
- Configured an OAuth 2.0 credential token containing a client ID and client secret.

**In your AWS account, make sure you have:**

- Created an [IAM role](#) for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your Microsoft Yammer authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

**Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).
Using the console

On the Yammer page, enter the following information:

1. **Name** – Name your data source for easy tracking.

   **Note**: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   **Note**: Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

3. **Authentication** – Choose between **New** and **Existing**.

   - If you choose **Existing**, select an existing secret for **Select secret**.

     If you choose **New**, enter the following information in the **New AWS Secrets Manager secret** section:

     - **Secret name** – A name for your secret.
     - **Username** – The username for your Microsoft Yammer Active Directory account.
     - **Password** – The password for your Microsoft Yammer Active Directory account.
     - **Client ID** – The OAuth client ID credential values you copied from your Microsoft Yammer account.
     - **Client secret** – The client secret from your Microsoft Yammer account.

4. **Configure VPC and security group** – **optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.
For more information, see VPC.

5. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see Identity crawler.

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content. For more information, see IAM role.

7. For **Sync scope**, provide the following information:
   - **sinceDate** – Select the date in your data source content from when Amazon Q should begin to crawl your data.
   - **Select content to sync** – Choose between All, Public messages, Attachments, and Inbox private messages.

8. For **Additional configuration – optional**, provide the following information:
   - **Community names** – Enter the community names you wish to include in your application.
   - **Regex patterns** – Add regular expression patterns to include or exclude certain file types. You can add up to 100 patterns.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.
   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New or modified content sync** – Sync only new and modified documents.
   - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

For more details, see Sync mode.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see Sync run schedule.

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See Tags for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the Data source details page after you finish adding your data source. You can choose from two types of fields:
a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.

You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

**Note**

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

### Using the API

You use the **CreateDataSource** action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**Yammer JSON schema**

The following is the Yammer JSON schema:

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "connectionConfiguration": {
            "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", "dateFieldFormat"]
            }]
          }
        }
      }
    }
  }
}
"indexFieldMapping": {
  "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": [
  "indexFieldMapping",
  "indexFieldDetail",
  "dataSourceFieldDetail"
]
"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"
 ]
}
"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": ["indexFieldName", "indexFieldType", "dataSourceFieldName"]
}

,
"required": [
  "fieldMappings"
]
,
"additionalProperties": {
  "type": "object",
  "properties": {
    "isCrawlAcl": {
      "type": "boolean"
    },
    "fieldForUserId": {
      "type": "string"
    },
    "inclusionPatterns": {
      "type": "array"
    },
    "exclusionPatterns": {
      "type": "array"
    },
    "sinceDate": {
      "type": "string",
      "pattern": "^(19|2[0-9]|3[01])T([0-9][0-9][0-9][0-9][0-9][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",
  "sinceDate"]
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source. This data source doesn't specify an endpoint in repositoryEndpointMetadata. Rather, the connection information is included in an AWS Secrets Manager secret that you provide the secretArn.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• community</td>
<td>A list of objects that map attributes or field names of Microsoft Yammer objects to Amazon Q index field names.</td>
</tr>
<tr>
<td>• user</td>
<td></td>
</tr>
<tr>
<td>• message</td>
<td></td>
</tr>
<tr>
<td>• attachment</td>
<td></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 Yammer data source. This includes your client ID and client secret.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>isCrawlMessage</td>
<td>Input TRUE to index</td>
</tr>
<tr>
<td>isCrawlAttachment</td>
<td></td>
</tr>
<tr>
<td>isCrawlPrivateMessage</td>
<td></td>
</tr>
<tr>
<td>sinceDate</td>
<td>Use to specify the time from when Amazon Q should crawl your Microsoft Yammer content</td>
</tr>
<tr>
<td>communityNameFilter</td>
<td>Use to specify community names to index.</td>
</tr>
<tr>
<td>inclusionPatterns</td>
<td>A list of regular expression patterns to include specific files in your 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. 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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>exclusionPatterns</td>
<td>A list of regular expression patterns to exclude specific files in your Yammer data source. Files that match the patterns are excluded from the files in your 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.</td>
</tr>
<tr>
<td>type</td>
<td>Specify YAMMER as your data source type</td>
</tr>
<tr>
<td>useChangeLog</td>
<td>true to use the Yammer change log to determine which documents require adding, updating, or deleting in the index.</td>
</tr>
</tbody>
</table>
| syncMode           | Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:  
  - Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index  
  - Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index  
  - Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index |
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enableIdentityCrawler</td>
<td>true to activate identity crawler. Identity crawler is activated by default. Crawling identity information on users and groups with access to certain documents is useful for user context filtering. Search results are filtered based on the user or their group access to documents. See <a href="#">Identity crawler</a> for more information.</td>
</tr>
</tbody>
</table>

**ACL crawling**

When you connect an Microsoft Yammer data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Microsoft Yammer instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

The group and user IDs are mapped as follows:

- **_email_id** – Your Microsoft email ID is an identifier that's necessary to configure each connector instance. Your email ID can be found in the properties section of your Microsoft account dashboard.

- **_group_id** – Group IDs exist in Microsoft Yammer Instances where there are set access permissions. They're mapped from the names of the groups in Microsoft Yammer.

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:
• Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.

• Permission to access the User Store API operations to ingest user and group access control information from documents.

• Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.

• (Optional) If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToGetSecret",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:{region}:{account_id}:secret:[[secret_id]]"
      ]
    },
    {
      "Sid": "AllowsAmazonQToDecryptSecret",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:{region}:{account_id}:key:[[key_id]]"
      ],
      "Condition": {
        "StringLike": {
          "kms:ViaService": [
            "secretsmanager.*.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AllowsAmazonQToIngestDocuments",
      "Effect": "Allow",
```
"Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
],
"Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},
{
    "Sid": "AllowsAmazonQToIngestPrincipalMapping",
    "Effect": "Allow",
    "Action": [
        "qbusiness:PutGroup",
        "qbusiness:CreateUser",
        "qbusiness:DeleteGroup",
        "qbusiness:UpdateUser",
        "qbusiness:ListGroups"
    ],
    "Resource": [
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/*"
    ]
},
{
    "Sid": "AllowsAmazonQToCreateAndDeleteNI",
    "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]]"
    ]
},
{
    "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface"
    ]
}
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
  "StringLike": {
    "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
  },
  "ForAllValues:StringEquals": {
    "aws:TagKeys": ["AMAZON_Q"
  ]
}
},
{
  "Sid": "AllowsAmazonQToCreateTags",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
    }
  }
},
{
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    }
  }
},
{
  "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeAvailabilityZones",
  ]
To allow Amazon Q to assume a role, you must also use the following trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Sid": "AllowsAmazonQServicePrincipal",
    "Effect": "Allow",
    "Principal": {
      "Service": "qbusiness.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "{{source_account}}"
      },
      "ArnEquals": {
        "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
      }
    }
  }
  ]
}
```

For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.
Connecting MySQL to Amazon Q

MySQL is an open source relational database management system. You can connect your MySQL instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

The Amazon Q MySQL data source connector supports MySQL 8.0. 21.

Learn more

• For an overview of the Amazon Q web experience creation process, see Configuring an application.
• For an overview of connector features, see Data source connector concepts.
• For information about connector configuration best practices, see Connector configuration best practices.

Topics

• Prerequisites
• Using the console
• Using the API
• ACL crawling
• IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In MySQL, make sure you have:

• Noted your database user name and password.

⚠️ Important

As a best practice, provide Amazon Q with read-only database credentials.

• Copied your database host URL, port, and instance.
In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your MySQL authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

**Note**

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

**Using the console**

On the MySQL page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In **Source**, enter the following information:
   
   a. **Host** – Enter the database host name.
   
   b. **Port** – Enter the database port.
   
   c. **Instance** – Enter the database instance.
   
   d. **Enable SSL certificate location** – Choose to enter the Amazon S3 path to your SSL certificate file.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.
4. In **Authentication** – Enter the following information for your **AWS Secrets Manager secret**.
   a. **Secret name** – A name for your secret.
   b. For **Database user name**, and **Password** – Enter the authentication credential values you copied from your database.
   c. Choose **Save**.

5. **Configure VPC and security group** – optional – Choose whether you want to use a VPC. If you do, enter the following information:
   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see **VPC**.

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see **IAM role**.

7. In **Sync scope**, enter the following information:
   - **SQL query** – Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Q 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 names of the document body column within your database table.

8. **In Additional configuration – optional** – Configure the following settings:

   • **Change-detecting columns** – Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q 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 Q 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.

9. **In Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   • **Full sync** – Sync all content regardless of the previous sync status.
   
   • **New or modified content sync** – Sync only new and modified documents.
   
   • **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

   For more details, see [Sync mode](##).

10. **In Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](##).

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](##) for more details.
12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

   **Note**

   You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

   **Using the API**

   You use the [CreateDataSource](http://example.com) action to connect a data source to your Amazon Q application.

   Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

   **JSON schema**

   The following is the JSON schema:

   ```json
   {
     "$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"
},
"serverlessAurora": {  
  "type": "string",
  "enum": ["true", "false"]
},
"required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
},
"type": {
  "type": "string",
  "pattern": "JDBC"
}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>Required configuration information for connecting your data source.</td>
</tr>
<tr>
<td></td>
<td>• dbType—The type of Java database you are using, whether mysql, db2, postgresql, oracle, or sqlserver.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>dbHost</td>
<td>The database host name.</td>
</tr>
<tr>
<td>dbPort</td>
<td>The database port.</td>
</tr>
<tr>
<td>dbInstance</td>
<td>The database instance.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>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.</td>
</tr>
<tr>
<td>document</td>
<td>A list of objects that map the attributes or field names of your database content to Amazon Q index field names. For more information, see <a href="#">Mapping data source fields</a>.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.</td>
</tr>
<tr>
<td>primaryKey</td>
<td>Provide the primary key for the database table. This identifies a table within your database.</td>
</tr>
<tr>
<td>titleColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>bodyColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>sqlQuery</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>timestampColumn</td>
<td>Enter the name of the column which contains time stamps. Amazon Q uses time stamp information to detect changes in your content and sync only changed content.</td>
</tr>
<tr>
<td>timestampFormat</td>
<td>Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.</td>
</tr>
<tr>
<td>timezone</td>
<td>Enter the name of the column which contains time zones for the content to be crawled.</td>
</tr>
<tr>
<td>changeDetectingColumns</td>
<td>Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q will re-index content when there is a change in any of these columns</td>
</tr>
<tr>
<td>allowedUsersColumns</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>allowedGroupsColumn</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>sourceURIColumn</td>
<td>Enter the name of the column which contains Source URLs to be indexed.</td>
</tr>
<tr>
<td>isSslEnabled</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify JDBC as your data source type.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
</table>
| syncMode      | Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose  
  - FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index  
  - FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index  
  - CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index. |
| 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:  
  ```json  
  {  
    "username": "database user name",  
    "password": "password"  
  }  
  ``` |
| version       | The version of the template that is currently supported. |

### ACL crawling

When you connect a database data source to Amazon Q, Amazon Q crawls user and group information from a column in the source table. You specify this column in the console or using the configuration parameter as part of the CreateDataSource operation.
If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- Authorization
- Identity crawler
- Understanding User Store

IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- Permission to access the SSL certificate stored in your Amazon S3 bucket.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToGetS3Objects",
            "Action": [
                "s3:GetObject"
            ]
        }
    ]
}
```
"Resource": [
    "arn:aws:s3:::{{input_bucket_name}}/*"
],
"Effect": "Allow",
"Condition": {
    "StringEquals": {
        "aws:ResourceAccount": "{{account_id}}"
    }
}
],
{
    "Sid": "AllowsAmazonQToGetSecret",
    "Effect": "Allow",
    "Action": [
        "secretsmanager:GetSecretValue"
    ],
    "Resource": [
        "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:{{secret_id}}"
    ]
}
],
{
    "Sid": "AllowsAmazonQToDecryptSecret",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:{{region}}:{{account_id}}:key:{{key_id}}"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
                "secretsmanager.*.amazonaws.com"
            ]
        }
    }
}
],
{
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
    ],
}
}
"Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}",

"Sid": "AllowsAmazonQToIngestPrincipalMapping",
"Effect": "Allow",
"Action": [
  "qbusiness:PutGroup",
  "qbusiness:CreateUser",
  "qbusiness:DeleteGroup",
  "qbusiness:UpdateUser",
  "qbusiness:ListGroups"
],
"Resource": [
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**
]
},

{"Sid": "AllowsAmazonQToCreateAndDeleteNI",
"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]]"
]
},

{"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
  "ec2:CreateNetworkInterface",
  "ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {"StringLike": {"aws:tag:TagKey": "TagValue"}}
}
"StringLike": {  
  "aws:RequestTag/AMAZON_Q":  
  "qbusiness_{{account_id}}_{{application_id}}_**"  
},  
  "ForAllValues:StringEquals": {  
    "aws:TagKeys": [  
      "AMAZON_Q"  
    ]  
  }  
},  
  "Sid": "AllowsAmazonQToCreateTags",  
  "Effect": "Allow",  
  "Action": [  
    "ec2:CreateTags"  
  ],  
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",  
  "Condition": {  
    "StringEquals": {  
      "ec2:CreateAction": "CreateNetworkInterface"  
    }  
  }  
},  
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",  
  "Effect": "Allow",  
  "Action": [  
    "ec2:CreateNetworkInterfacePermission"  
  ],  
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",  
  "Condition": {  
    "StringLike": {  
      "aws:ResourceTag/AMAZON_Q":  
      "qbusiness_{{account_id}}_{{application_id}}_**"  
    }  
  }  
},  
  "Sid": "AllowsAmazonQToDescribeResourcesForVPC",  
  "Effect": "Allow",  
  "Action": [  
    "ec2:DescribeNetworkInterfaces",  
    "ec2:DescribeAvailabilityZones",  
    "ec2:DescribeVpcs"  
  ],  
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:vpc/**",  
  "Condition": {  
    "StringLike": {  
      "aws:RequestTag/AMAZON_Q":  
      "qbusiness_{{account_id}}_{{application_id}}_**"  
    }  
  }  
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
        }
      }
    }
  ]
}
```

For more information on Amazon Q data source connector IAM roles, see [IAM roles for Amazon Q data source connectors](#).
Connecting Oracle Database to Amazon Q

Oracle Database is a database management system. You can connect your Oracle Database instance to Amazon Q—using either the AWS Management Console, CLI, or the `CreateDataSource` API—and create an Amazon Q web experience.

The Amazon Q Oracle Database data source connector supports Oracle Database 18c, 19c, and 21c.

Learn more

- For an overview of the Amazon Q web experience creation process, see [Configuring an application](#).
- For an overview of connector features, see [Data source connector concepts](#).
- For information about connector configuration best practices, see [Connector configuration best practices](#).

Topics

- [Prerequisites](#)
- [Using the console](#)
- [Using the API](#)
- [ACL crawling](#)
- [IAM roles](#)

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

**In Oracle Database, make sure you have:**

- Noted your database user name and password.

⚠️ **Important**

As a best practice, provide Amazon Q with read-only database credentials.

- Copied your database host URL, port, and instance.
In your AWS account, make sure you have:

- Created an **IAM role** for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your Oracle Database authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

**Note**
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

**Using the console**

On the **Oracle Database** page, enter the following information:

1. **Name** – Name your data source for easy tracking.

   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In **Source**, enter the following information:

   a. **Host** – Enter the database host name.
   b. **Port** – Enter the database port.
   c. **Instance** – Enter the database instance.
   d. **Enable SSL certificate location** – Choose to enter the Amazon S3 path to your SSL certificate file.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.
Note

Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).

4. In Authentication – Enter the following information for your AWS Secrets Manager secret.
   a. **Secret name** – A name for your secret.
   b. For **Database user name**, and **Password** – Enter the authentication credential values you copied from your database.
   c. Choose **Save**.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:
   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see **VPC**.

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see **IAM role**.

7. In **Sync scope**, enter the following information:
   - **SQL query** – Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Q 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 names of the document body column within your database table.

8. **In Additional configuration – optional** – Configure the following settings:

• **Change-detecting columns** – Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q 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 Q 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.

9. **In Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

• **Full sync** – Sync all content regardless of the previous sync status.

• **New or modified content sync** – Sync only new and modified documents.

• **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

For more details, see [Sync mode](#).

10. **In Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.
12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

   ![Note]

   You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**JSON schema**

The following is the JSON schema:

```json
{
   "$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"
      ]
    ]
  }
},
"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"
},
"serverlessAurora": {
    "type": "string",
    "enum": ["true", "false"]
},
"required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
},
"type": {
    "type": "string",
    "pattern": "JDBC"}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>Required configuration information for connecting your data source.</td>
</tr>
<tr>
<td></td>
<td>• dbType—The type of Java database you are using, whether mysql, db2,</td>
</tr>
<tr>
<td></td>
<td>postgresql, oracle, or sqlserver.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
|                       | **dbHost**—The database host name.  
|                       | **dbPort**—The database port.  
<p>|                       | <strong>dbInstance</strong>—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 Q index field names. For more information, see <a href="#">Mapping data source fields</a>. |
| 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 Q will crawl all database content that matches your query. |</p>
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestampColumn</td>
<td>Enter the name of the column which contains time stamps. Amazon Q uses time stamp information to detect changes in your content and sync only changed content.</td>
</tr>
<tr>
<td>timestampFormat</td>
<td>Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.</td>
</tr>
<tr>
<td>timezone</td>
<td>Enter the name of the column which contains time zones for the content to be crawled.</td>
</tr>
<tr>
<td>changeDetectingColumns</td>
<td>Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q will re-index content when there is a change in any of these columns</td>
</tr>
<tr>
<td>allowedUsersColumns</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>allowedGroupsColumn</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>sourceURIColumn</td>
<td>Enter the name of the column which contains Source URLs to be indexed.</td>
</tr>
<tr>
<td>isSslEnabled</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify JDBC as your data source type.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
</table>
| syncMode      | Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose  
  - FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index  
  - FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index  
  - CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index. |
| 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. |

### ACL crawling

When you connect a database data source to Amazon Q, Amazon Q crawls user and group information from a column in the source table. You specify this column in the console or using the configuration parameter as part of the CreateDataSource operation.
If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- Authorization
- Identity crawler
- Understanding User Store

IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- Permission to access the SSL certificate stored in your Amazon S3 bucket.
- (Optional) If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowsAmazonQToGetS3Objects",
        "Action": [
            "s3:GetObject"
        ],
```
"Resource": [
  "arn:aws:s3:::{input_bucket_name}/*"
],
"Effect": "Allow",
"Condition": {
  "StringEquals": {
    "aws:ResourceAccount": "{{account_id}}"
  }
}
},
{
  "Sid": "AllowsAmazonQToGetSecret",
  "Effect": "Allow",
  "Action": [
    "secretsmanager:GetSecretValue"
  ],
  "Resource": [
    "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
  ]
},
{
  "Sid": "AllowsAmazonQToDecryptSecret",
  "Effect": "Allow",
  "Action": [
    "kms:Decrypt"
  ],
  "Resource": [
    "arn:aws:kms:{{region}}:{{account_id}}:key:[[key_id]]"
  ],
  "Condition": {
    "StringLike": {
      "kms:ViaService": [
        "secretsmanager.*.amazonaws.com"
      ]
    }
  }
},
{
  "Sid": "AllowsAmazonQToIngestDocuments",
  "Effect": "Allow",
  "Action": [
    "qbusiness:BatchPutDocument",
    "qbusiness:BatchDeleteDocument"
  ]
}
"Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}",
{
"Sid": "AllowsAmazonQToIngestPrincipalMapping",
"Effect": "Allow",
"Action": [
"qbusiness:PutGroup",
"qbusiness:CreateUser",
"qbusiness:DeleteGroup",
"qbusiness:UpdateUser",
"qbusiness:ListGroups"
],
"Resource": [
"arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
"arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
"arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/*"
]
},
{
"Sid": "AllowsAmazonQToCreateAndDeleteNI",
"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]]"
]
},
{
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterface",
"ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {

"StringLike": {
    "aws:RequestTag/AMAZON_Q": "qbusiness_{{{account_id}}}{{{application_id}}}_*"
},
"ForAllValues:StringEquals": {
    "aws:TagKeys": [
        "AMAZON_Q"
    ]
}
}
{
    "Sid": "AllowsAmazonQToCreateTags",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "CreateNetworkInterface"
        }
    }
}
{
    "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterfacePermission"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
    "Condition": {
        "StringLike": {
            "aws:ResourceTag/AMAZON_Q": "qbusiness_{{{account_id}}}{{{application_id}}}_*"
        }
    }
}
{  
    "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeAvailabilityZones",
    ]
}
"ec2:DescribeNetworkInterfaceAttribute",
"ec2:DescribeVpcs",
"ec2:DescribeRegions",
"ec2:DescribeNetworkInterfacePermissions",
"ec2:DescribeSubnets"
],
"Resource": "*"
}
]
}

To allow Amazon Q to assume a role, you must also use the following trust policy:

{
 "Version": "2012-10-17",
 "Statement": [
 {  
 "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
 "Effect": "Allow",
 "Principal": {
 "Service": "qbusiness.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "aws:SourceAccount": "{{source_account}}"
 },
 "ArnLike": {
 "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
 }
 }
 ]
}

For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.
Connecting PostgreSQL to Amazon Q

PostgreSQL is an open source database management system. You can connect your PostgreSQL instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

The Amazon Q PostgreSQL data source connector supports PostgreSQL 9.6.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In PostgreSQL, make sure you have:

- Noted your database user name and password.

⚠️ Important

As a best practice, provide Amazon Q with read-only database credentials.

- Copied your database host URL, port, and instance.
In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your PostgreSQL authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note: If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the PostgreSQL page, enter the following information:

1. Name – Name your data source for easy tracking.
   
   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In Source, enter the following information:
   
   a. Host – Enter the database host URL.
   b. Port – Enter the database port, for example, 5432.
   c. Instance – Enter the database instance, for example postgres.
   d. Enable SSL certificate location – Choose to enter the Amazon S3 path to your SSL certificate file.

3. Authorization – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.
4. In **Authentication** – Enter the following information for your **AWS Secrets Manager secret**.
   a. **Secret name** – A name for your secret.
   b. For **Database user name**, and **Password** – Enter the authentication credential values you copied from your database.
   c. Choose **Save**.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:
   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

7. In **Sync scope**, enter the following information:
   - **SQL query** – Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB and not contain any semi-colons (;). Amazon Q 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 names of the document body column within your database table.

8. In **Additional configuration – optional** – Configure the following settings:

- **Change-detecting columns** – Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q 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 Q 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.

9. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

- **Full sync** – Sync all content regardless of the previous sync status.

- **New or modified content sync** – Sync only new and modified documents.

- **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

For more details, see **Sync mode**.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see **Sync run schedule**.

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See **Tags** for more details.
12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the Data source details page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see Field mappings.

13. To finish connecting your data source to Amazon Q, select Add data source. 

   You are taken to the Data source details, where you can view your data source configuration details.

14. In Data source details, choose Sync now to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

    **Note**

    You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**JSON schema**

The following is the JSON schema:

```json
{
   "$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": "string"
    }
  }
}
"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": [
      "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"
},
"serverlessAurora": {
    "type": "string",
    "enum": ["true", "false"]
},
"required": ["primaryKey", "titleColumn", "bodyColumn", "sqlQuery"]
},
"type": {
    "type": "string",
    "pattern": "JDBC"}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>Required configuration information for connecting your data source.</td>
</tr>
<tr>
<td></td>
<td>• dbType—The type of Java database you are using, whether mysql, db2, postgresql, oracle, or sqlserver.</td>
</tr>
</tbody>
</table>

PostgreSQL
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbHost</td>
<td>The database host name.</td>
</tr>
<tr>
<td>dbPort</td>
<td>The database port.</td>
</tr>
<tr>
<td>dbInstance</td>
<td>The database instance.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>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.</td>
</tr>
<tr>
<td>document</td>
<td>A list of objects that map the attributes or field names of your database content to Amazon Q index field names. For more information, see <a href="#">Mapping data source fields</a>.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source. Use to include or exclude specific content in your database data source.</td>
</tr>
<tr>
<td>primaryKey</td>
<td>Provide the primary key for the database table. This identifies a table within your database.</td>
</tr>
<tr>
<td>titleColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>bodyColumn</td>
<td>Provide the name of the document title column within your database table.</td>
</tr>
<tr>
<td>sqlQuery</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>timestampColumn</td>
<td>Enter the name of the column which contains time stamps. Amazon Q uses time stamp information to detect changes in your content and sync only changed content.</td>
</tr>
<tr>
<td>timestampFormat</td>
<td>Enter the name of the column which contains time stamp formats to use to detect content changes and re-sync your content.</td>
</tr>
<tr>
<td>timezone</td>
<td>Enter the name of the column which contains time zones for the content to be crawled.</td>
</tr>
<tr>
<td>changeDetectingColumns</td>
<td>Enter the names of the columns that Amazon Q will use to detect content changes. Amazon Q will re-index content when there is a change in any of these columns</td>
</tr>
<tr>
<td>allowedUsersColumns</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>allowedGroupsColumn</td>
<td>Enter the name of the column which contains User IDs to be allowed access to content.</td>
</tr>
<tr>
<td>sourceURIColumn</td>
<td>Enter the name of the column which contains Source URLs to be indexed.</td>
</tr>
<tr>
<td>isSslEnabled</td>
<td>Enter SQL query statements like SELECT and JOIN operations. SQL queries must be less than 32KB. Amazon Q will crawl all database content that matches your query.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify JDBC as your data source type.</td>
</tr>
</tbody>
</table>
## Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
</table>
| syncMode      | Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose  
- **FORCED_FULL_CRAWL** to freshly re-crawl all content and replace existing content each time your data source syncs with your index  
- **FULL_CRAWL** to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index  
- **CHANGE_LOG** to incrementally crawl only new and modified content each time your data source syncs with your index. |
| 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:  
```json  
{  
  "user name": "database user name",  
  "password": "password"  
}  
``` |
| version       | The version of the template that is currently supported. |

### ACL crawling

When you connect a database data source to Amazon Q, Amazon Q crawls user and group information from a column in the source table. You specify this column in the console or using the configuration parameter as part of the CreateDataSource operation.
If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- Authorization
- Identity crawler
- Understanding User Store

IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- Permission to access the SSL certificate stored in your Amazon S3 bucket.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowsAmazonQToGetS3Objects",
    "Action": [
      "s3:GetObject"
    ],
```
"Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}",

    
    
    "Sid": "AllowsAmazonQToIngestPrincipalMapping",
    "Effect": "Allow",
    "Action": [
        "qbusiness:PutGroup",
        "qbusiness:CreateUser",
        "qbusiness:DeleteGroup",
        "qbusiness:UpdateUser",
        "qbusiness:ListGroups"
    ],
    "Resource": [
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
        "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**"
    ]
},

    
    
    "Sid": "AllowsAmazonQToCreateAndDeleteNI",
    "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]]"
    ]
},

    
    
    "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
    "Condition": {
"StringLike": {
  "aws:RequestTag/AMAZON_Q":
  "qbusiness_{{account_id}}_{{application_id}}_*"
},
"ForAllValues:StringEquals": {
  "aws:TagKeys": [
    "AMAZON_Q"
  ]
}
},
{
  "Sid": "AllowsAmazonQToCreateTags",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
    }
  }
},
{
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/AMAZON_Q":
      "qbusiness_{{account_id}}_{{application_id}}_*"
    }
  }
},
{
  "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeNetworkInterfacePermissions"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
    }
  }
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAmazonQToAssumeRoleForServicePrincipal",
            "Effect": "Allow",
            "Principal": {
                "Service": "qbusiness.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "{{source_account}}"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
                }
            }
        }
    ]
}
```

For more information on Amazon Q data source connector IAM roles, see [IAM roles for Amazon Q data source connectors](https://aws.amazon.com/q/business/).
Connecting Quip to Amazon Q

Quip is a collaborative productivity software that offers real time document-authoring capabilities. You can connect your Quip instance to Amazon Q—using either the AWS Management Console, CLI, or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Quip, make sure you have:

- A Quip account with administrative permissions.
- Created Quip authentication credentials that include a personal access token. See Quip documentation on authentication for more information.
- Copied your Quip site domain. For example, https://quip-company.quivdomain.com/browse where quipdomain is the domain.

In your AWS account, make sure you have:
• Created an [IAM role](#) for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

• Stored your Quip authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

![Note]

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see [Data source connector configuration best practices](#).

**Using the console**

On the **Quip** page, enter the following information:

1. **Name** – Name your data source for easy tracking.

   ![Note]

   You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumerical characters.

2. **Source** – Enter your **Quip domain name**. You can find your domain name in the browser URL of your Quip. For example, `https://quip-company.quipdomain.com/browse`, the domain is "quipdomain".

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   ![Note]

   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

4. **Authentication** – Enter the following information for your **AWS Secrets Manager secret**.

   a. **Secret name** – A name for your secret.

   b. **Quip token** – Enter the Quip personal access token you created in your Quip account.
5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

7. In **Sync scope**, enter the following information:

   a. **Add Quip folder IDs to crawl** – Enter the Quip folder IDs you want to crawl. You can find your folder ID in the browser URL when you access your folder in Quip. For example, `https://quip-company.quipdomain.com/zlLuOVNSarTL/folder-name`, the folder ID is "zlLuOVNSarTL"..

   b. (Optional) **Additional configuration – optional** – Configure the following settings:

      - **Content types** – Choose between crawling **All content**, **File comments**, **Chat rooms** and **Attachments**.

      - **Regex patterns** – Add regex patterns to include or exclude file names, file types, or file paths. You can have a total of 100 patterns.

8. 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 Q for the first time, all content is synced by default.

    - **Full sync** – Sync all content regardless of the previous sync status.

    - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

9. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

10. **Tags – optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.
11. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can’t edit these.

b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

For more information, see **Field mappings**.

12. To finish connecting your data source to Amazon Q, select **Add data source**.

You are taken to the **Data source details**, where you can view your data source configuration details.

13. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

**Note**

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

### Using the API

You use the **CreateDataSource** action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**Quip JSON schema**

The following is the Quip JSON schema:

```json
{
  "$schema": "http://json-schema.org/draft-04/schema#",
```
"type": "object",
"properties": {
  "connectionConfiguration": {
    "type": "object",
    "properties": {
      "repositoryEndpointMetadata": {
        "type": "object",
        "properties": {
          "domain": {
            "type": "string"
          }
        },
        "required": [
          "domain"
        ]
      },
      "required": [
        "repositoryEndpointMetadata"
      ]
    },
    "repositoryConfigurations": {
      "type": "object",
      "properties": {
        "thread": {
          "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"
]
}
"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" }
} }
} } }
"dateFormat": {
  "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"
            },
            "dateFormat": {
              "type": "string",
              "pattern": "yyyy-MM-dd'T'HH:mm:ss'Z'"
"required": [
  "indexFieldName",
  "indexFieldType",
  "dataSourceFieldName"
],
"additionalProperties": {
  "type": "object",
  "properties": {
    "isCrawlAcl": {
      "type": "boolean"
    },
    "fieldForUserId": {
      "type": "string"
    },
    "folderIds": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "crawlFileComments": {
      "type": "boolean"
    },
    "crawlChatRooms": {
      "type": "boolean"
    },
    "crawlAttachments": {
      "type": "boolean"
    },
    "inclusionPatterns": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
}
"type": "string"
],
"required": []
},
"type": {
  "type": "string",
  "pattern": "QUIP"
},
"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"
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>domain</td>
<td>Your Quip site domain. For example, <a href="https://quip-company.quipdomain.main.com/browse">https://quip-company.quipdomain.main.com/browse</a> where quipdomain is the domain.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• thread</td>
<td>A list of objects that map the attributes or field names of your Quip pages and assets to Amazon Q index field names.</td>
</tr>
<tr>
<td>• message</td>
<td></td>
</tr>
<tr>
<td>• attachment</td>
<td></td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>• crawlFileComments</td>
<td>true to index.</td>
</tr>
<tr>
<td>• crawlChatRooms</td>
<td></td>
</tr>
<tr>
<td>• crawlAttachments</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• inclusionPatterns</td>
<td>A list of regular expression patterns to include specific content in your Quip data source. Content that matches the patterns are included in the index. Content that doesn't match the pattern 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.</td>
</tr>
<tr>
<td>• exclusionPatterns</td>
<td>A list of regular expression patterns to exclude specific content in your Quip 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.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify QUIP as your data source type.</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>Specify true to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to specific documents.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>- Use <code>FORCED_FULL_CRAWL</code> to freshly re-crawl all content and replace existing content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>- Use <code>FULL_CRAWL</code> to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index.</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 Quip. The secret must contain a JSON structure with the following keys:</td>
</tr>
</tbody>
</table>
|               | ```
|               |    
|               |    
|               |    
|               |    |
|               |   |
|               | ``` |
| version       | The version of this template that's currently supported. |

### ACL crawling

When you connect an Quip data source to AWS Enterprise !, AWS Enterprise ! crawls ACL information attached to a document (user and group information) from your Quip instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user’s document access level.

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.

For more information, see:

• Authorization
• Identity crawler
• Understanding User Store

IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

• Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
• Permission to access the User Store API operations to ingest user and group access control information from documents.
• Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
• (Optional) If you’re using Amazon VPC, permission to access your Amazon VPC.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowsAmazonQToGetSecret",
         "Effect": "Allow",
         "Action": [
            "secretsmanager:GetSecretValue"
         ],
         "Resource": [
            "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
         ]
      },
      {
         "Sid": "AllowsAmazonQToDecryptSecret",
         "Effect": "Allow",
         "Action": [
            "secretsmanager:Decrypt"
         ],
         "Resource": [
            "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
         ]
      }
   ]
}
```
"Action": [  "kms:Decrypt"
],  "Resource": [  "arn:aws:kms:{{region}}:{{account_id}}:key/[[[key_id]]]"
],  "Condition": {  "StringLike": {  "kms:ViaService": [  "secretsmanager.*.amazonaws.com"
  ]
  }
  }
],
{
  "Sid": "AllowsAmazonQToIngestDocuments",
  "Effect": "Allow",
  "Action": [  "qbusiness:BatchPutDocument",
  "qbusiness:BatchDeleteDocument"
  ],
  "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
},
{
  "Sid": "AllowsAmazonQToIngestPrincipalMapping",
  "Effect": "Allow",
  "Action": [  "qbusiness:PutGroup",
  "qbusiness:CreateUser",
  "qbusiness:DeleteGroup",
  "qbusiness:UpdateUser",
  "qbusiness:ListGroups"
  ],
  "Resource": [  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**
  ]
},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",  "Effect": "Allow",
  "Action": [  "qbusiness:CreateIndex",
  "qbusiness:DeleteIndex"
  ],
  "Resource": [  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}"
  ]
}
"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]]"
]
},
{
   "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
   "Effect": "Allow",
   "Action": [
      "ec2:CreateNetworkInterface",
      "ec2:DeleteNetworkInterface"
   ],
   "Resource": "arn:aws:ec2:{region}:{account_id}:network-interface/**",
   "Condition": {
      "StringLike": {
         "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
      },
      "ForAllValues:StringEquals": {
         "aws:TagKeys": [
            "AMAZON_Q"
         ]
      }
   }
},
{
   "Sid": "AllowsAmazonQToCreateTags",
   "Effect": "Allow",
   "Action": [
      "ec2:CreateTags"
   ],
   "Resource": "arn:aws:ec2:{region}:{account_id}:network-interface/**",
   "Condition": {
      "StringEquals": {
         "ec2:CreateAction": "CreateNetworkInterface"
      }
   }
},
{
   "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
   "Effect": "Allow",
   "Action": [
      "ec2:CreateNetworkInterface",
      "ec2:DeleteNetworkInterface"
   ],
   "Resource": "arn:aws:ec2:{region}:{account_id}:network-interface/**",
   "Condition": {
      "StringLike": {
         "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
      },
      "ForAllValues:StringEquals": {
         "aws:TagKeys": [
            "AMAZON_Q"
         ]
      }
   }
}
"Effect": "Allow",
"Action": [
  "ec2:CreateNetworkInterfacePermission"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
  "StringLike": {
    "aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
  }
}
},
{
  "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeNetworkInterfaceAttribute",
    "ec2:DescribeVpcs",
    "ec2:DescribeRegions",
    "ec2:DescribeNetworkInterfacePermissions",
    "ec2:DescribeSubnets"
  ],
  "Resource": "*"
}
}

To allow Amazon Q to assume a role, you must also use the following trust policy:

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        }
      }
    }
  ]
}
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Connecting Salesforce Online to Amazon Q

Salesforce is a customer relationship management (CRM) tool for managing support, sales, and marketing teams. You can connect Salesforce instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

Learn more

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.
In Salesforce, make sure you have:

- Copied the Salesforce security token associated with the account that’s 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. For more information, see Salesforce documentation on Connected Apps on the Salesforce website.
- 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 Q connects to.

In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your Salesforce authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Salesforce Online page, enter the following information:

1. Name – Name your data source for easy tracking.

   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.
2. In **Source**, enter the following information:

   - **Salesforce URL** – Enter your Salesforce server URL. For example, `https://mysite.salesforce.com`.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See [Authorization](#) for more details.

   ![Note]
   
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see [Integrating with an Identity Provider (IdP)](#).

4. **Authentication** – Enter the following information for your **AWS Secrets Manager secret**.

   a. **Secret name** – A name for your secret.
   
   b. For **Username, Password, Security token, Consumer key, Consumer secret, and Authentication URL** – Enter the authentication credential values that you created in your Salesforce account.
   
   c. Choose **Save and add secret**.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
   
   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see [VPC](#).

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).
8. **Sync scope** – Set the content that you want to sync.

   a. For **Standard objects**, **Standard objects with attachments**, **Standard objects without attachments**, and **Knowledge articles** – 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 up to 10 article types.

   b. For **Custom objects** – Add custom object names. You can choose to include custom object attachments as well.

9. In **Additional configuration** – **optional**:

   - For **Entity regex patterns** and **Attachment regex patterns** – Add regular expression patterns to include or exclude certain files. You can add up to 100 patterns.

10. In **Sync mode**, choose how you want to update your index when your data source content changes. When you sync your data source with Amazon Q for the first time, all content is synced by default.

   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New or modified content sync** – Sync only new and modified documents.
   - **New, modified, or deleted content sync** – Sync only new, modified, and deleted documents.

   For more details, see **Sync mode**.

11. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see **Sync run schedule**.

12. **Tags** – **optional** – Add tags to search and filter your resources or track your AWS costs. See **Tags** for more details.
13. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.

   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

14. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

15. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

   **Note**
   
   You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

### Using the API

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

#### Salesforce JSON schema

The following is the Salesforce JSON schema:

```json
{
    "$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"
          ]
        ]
      }
    },
    "attachment": {
      "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"
                ]
              ]
            }
          }
        }
      }
    }
  }
}
"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"
            ]
        ]
    },
    "serviceCatalog": {
        "type": "object",
        "properties": {
            "Salesforce Online": {
                "type": "object",
                "properties": {
                    "837": {
                        "type": "object",
                        "properties": {
                            "type": "string"
                        }
                    }
                }
            }
        }
    }
}
"required": [
    "fieldMappings"
],
"serviceCatalog": {
    "type": "object",
    "properties": {
        "Salesforce Online": {
            "type": "object",
            "properties": {
                "837": {
                    "type": "object",
                    "properties": {
                        "type": "string"
                    }
                }
            }
        }
    }
}
"fieldMappings": [
    
],
"required": [
    "fieldMappings"
]
},
"incident": {
    "type": "object",
    "properties": {
        "fieldMappings": {
            "type": "array",
            "items": [
        
    ],
    
],
"required": [
    "fieldMappings"
]
}

{
  "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"
  ]
}
}
"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"
  }
}
"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",
  "syncMode",
  "additionalProperties",
  "secretArn",
  "type"
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>hostUrl</td>
<td>The URL of the Salesforce instance to be indexed.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• account</td>
<td>A list of objects that map the attributes or field names of your Salesforce entities to Amazon Q index field names.</td>
</tr>
<tr>
<td>• contact</td>
<td></td>
</tr>
<tr>
<td>• campaign</td>
<td></td>
</tr>
<tr>
<td>• case</td>
<td></td>
</tr>
<tr>
<td>• product</td>
<td></td>
</tr>
<tr>
<td>• lead</td>
<td></td>
</tr>
<tr>
<td>• contract</td>
<td></td>
</tr>
<tr>
<td>• partner</td>
<td></td>
</tr>
<tr>
<td>• profile</td>
<td></td>
</tr>
<tr>
<td>• idea</td>
<td></td>
</tr>
<tr>
<td>• pricebook</td>
<td></td>
</tr>
<tr>
<td>• task</td>
<td></td>
</tr>
<tr>
<td>• solution</td>
<td></td>
</tr>
<tr>
<td>• attachment</td>
<td></td>
</tr>
<tr>
<td>• user</td>
<td></td>
</tr>
<tr>
<td>• document</td>
<td></td>
</tr>
<tr>
<td>• knowledgeArticles</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>group</td>
<td></td>
</tr>
<tr>
<td>opportunity</td>
<td></td>
</tr>
<tr>
<td>chatter</td>
<td></td>
</tr>
<tr>
<td>customEntity</td>
<td></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 Salesforce data source. The secret must contain a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;authenticationUrl&quot;: &quot;The OAUTH endpoint that Amazon Q connects to get an OAUTH token.&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;consumerKey&quot;: &quot;The application public key generated when you created your Salesforce application.&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;consumerSecret&quot;: &quot;The application private key generated when you created your Salesforce application.&quot; ,</td>
</tr>
<tr>
<td></td>
<td>&quot;password&quot;: &quot;The password associated with the user logging in to the Salesforce instance.&quot; ,</td>
</tr>
<tr>
<td></td>
<td>&quot;securityToken&quot;: &quot;The token associated with the user account logging in to the Salesforce instance.&quot; ,</td>
</tr>
<tr>
<td></td>
<td>&quot;username&quot;: &quot;The user name of the user logging in to the Salesforce instance.&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify <code>true</code> to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for <code>UserId</code> for ACL crawling.</td>
</tr>
<tr>
<td>accountFilter</td>
<td></td>
</tr>
<tr>
<td>contactFilter</td>
<td></td>
</tr>
<tr>
<td>caseFilter</td>
<td></td>
</tr>
<tr>
<td>campaignFilter</td>
<td></td>
</tr>
<tr>
<td>contractFilter</td>
<td></td>
</tr>
<tr>
<td>groupFilter</td>
<td></td>
</tr>
<tr>
<td>leadFilter</td>
<td></td>
</tr>
<tr>
<td>productFilter</td>
<td></td>
</tr>
<tr>
<td>opportunityFilter</td>
<td>A collection of strings that specifies which entities to filter.</td>
</tr>
<tr>
<td>partnerFilter</td>
<td></td>
</tr>
<tr>
<td>pricebookFilter</td>
<td></td>
</tr>
<tr>
<td>ideaFilter</td>
<td></td>
</tr>
<tr>
<td>profileFilter</td>
<td></td>
</tr>
<tr>
<td>taskFilter</td>
<td></td>
</tr>
<tr>
<td>solutionFilter</td>
<td></td>
</tr>
<tr>
<td>userFilter</td>
<td></td>
</tr>
<tr>
<td>chatterFilter</td>
<td></td>
</tr>
<tr>
<td>documentFilter</td>
<td></td>
</tr>
<tr>
<td>knowledgeArticleFilter</td>
<td></td>
</tr>
<tr>
<td>customEntities</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>inclusionPatterns</td>
<td>A list of regular expression patterns to <em>include</em> specific 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.</td>
</tr>
<tr>
<td>• inclusionDocumentFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionDocumentFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionAccountFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionCampaignFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionCampaignFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionCaseFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionCaseFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionContactFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionContactFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionContractFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionLeadFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionLeadFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionOpportunityFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionOpportunityFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionSolutionFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionSolutionFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>• inclusionTaskFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>inclusionTaskFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionGroupFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionGroupFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionChatterFileTypePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionChatterFileNamePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionCustomEntityTypePatterns</td>
<td></td>
</tr>
<tr>
<td>inclusionCustomEntityFileNamePatterns</td>
<td></td>
</tr>
</tbody>
</table>
## Configuration

<table>
<thead>
<tr>
<th>exclusionPatterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>- exclusionDocumentFileTypePatterns</td>
</tr>
<tr>
<td>- exclusionDocumentFileNamePatterns</td>
</tr>
<tr>
<td>- exclusionAccountFileTypePatterns</td>
</tr>
<tr>
<td>- exclusionCampaignFileTypePatterns</td>
</tr>
<tr>
<td>- exclusionCampaignFileNamePatterns</td>
</tr>
<tr>
<td>- exclusionCaseFileTypePatterns</td>
</tr>
<tr>
<td>- exclusionCaseFileNamePatterns</td>
</tr>
<tr>
<td>- exclusionContactFileTypePatterns</td>
</tr>
<tr>
<td>- exclusionContractFileNamePatterns</td>
</tr>
<tr>
<td>- exclusionLeadFileTypePatterns</td>
</tr>
<tr>
<td>- exclusionLeadFileNamePatterns</td>
</tr>
<tr>
<td>- exclusionOpportunityFileTypePatterns</td>
</tr>
<tr>
<td>- exclusionOpportunityFileNamePatterns</td>
</tr>
<tr>
<td>- exclusionSolutionFileTypePatterns</td>
</tr>
<tr>
<td>- exclusionSolutionFileNamePatterns</td>
</tr>
<tr>
<td>- exclusionTaskFileTypePatterns</td>
</tr>
<tr>
<td>- exclusionTaskFileNamePatterns</td>
</tr>
<tr>
<td>- exclusionGroupFileTypePatterns</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A list of regular expression patterns to exclude specific 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 precedence and the file isn't included in the index.</td>
</tr>
<tr>
<td>Configuration</td>
</tr>
<tr>
<td>-------------------------------------------------------------------</td>
</tr>
<tr>
<td>exclusionGroupFileNamePatterns</td>
</tr>
<tr>
<td>exclusionChatterFileTypePatterns</td>
</tr>
<tr>
<td>exclusionChatterFileNamePatterns</td>
</tr>
<tr>
<td>exclusionCustomEntityFileTypePatterns</td>
</tr>
<tr>
<td>exclusionCustomEntityFileNamePatterns</td>
</tr>
<tr>
<td>Configuration</td>
</tr>
<tr>
<td>---------------------------------------</td>
</tr>
<tr>
<td>isCrawlAccount</td>
</tr>
<tr>
<td>isCrawlContact</td>
</tr>
<tr>
<td>isCrawlCase</td>
</tr>
<tr>
<td>isCrawlCampaign</td>
</tr>
<tr>
<td>isCrawlProduct</td>
</tr>
<tr>
<td>isCrawlLead</td>
</tr>
<tr>
<td>isCrawlContract</td>
</tr>
<tr>
<td>isCrawlPartner</td>
</tr>
<tr>
<td>isCrawlProfile</td>
</tr>
<tr>
<td>isCrawlIdea</td>
</tr>
<tr>
<td>isCrawlPricebook</td>
</tr>
<tr>
<td>isCrawlDocument</td>
</tr>
<tr>
<td>crawlSharedDocument</td>
</tr>
<tr>
<td>isCrawlGroup</td>
</tr>
<tr>
<td>isCrawlOpportunity</td>
</tr>
<tr>
<td>isCrawlChatter</td>
</tr>
<tr>
<td>isCrawlUser</td>
</tr>
<tr>
<td>isCrawlSolution</td>
</tr>
<tr>
<td>isCrawlTask</td>
</tr>
<tr>
<td>isCrawlAccountAttachments</td>
</tr>
<tr>
<td>isCrawlContactAttachments</td>
</tr>
<tr>
<td>isCrawlCaseAttachments</td>
</tr>
<tr>
<td>isCrawlCampaignAttachments</td>
</tr>
<tr>
<td>isCrawlLeadAttachments</td>
</tr>
<tr>
<td>isCrawlContractAttachments</td>
</tr>
<tr>
<td>isCrawlGroupAttachments</td>
</tr>
<tr>
<td>isCrawlOpportunityAttachments</td>
</tr>
<tr>
<td>isCrawlChatterAttachments</td>
</tr>
<tr>
<td>Configuration</td>
</tr>
<tr>
<td>---------------------------------------</td>
</tr>
<tr>
<td>• isCrawlSolutionAttachments</td>
</tr>
<tr>
<td>• isCrawlTaskAttachments</td>
</tr>
<tr>
<td>• isCrawlCustomEntityAttachments</td>
</tr>
<tr>
<td>• isCrawlKnowledgeArticles</td>
</tr>
<tr>
<td>• isCrawlDraft</td>
</tr>
<tr>
<td>• isCrawlPublish</td>
</tr>
<tr>
<td>• isCrawlArchived</td>
</tr>
<tr>
<td>type</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>• Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index</td>
</tr>
<tr>
<td></td>
<td>• Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index</td>
</tr>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>

### ACL crawling

When you connect an Salesforce data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Salesforce instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

You can apply ACL based chat filtering using Salesforce standard objects and chatter feeds. ACL based chat filtering isn't available for Salesforce knowledge articles.

**For standard objects, the _user_id and _group_ids are used as follows:**

- _user_id – The username of the Salesforce user.
- _group_ids – The group names in Salesforce.
  - 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 username 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.

For more information, see:

• Authorization
• Identity crawler
• Understanding User Store

IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

• Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
• Permission to access the User Store API operations to ingest user and group access control information from documents.
• Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
• (Optional) If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
```
"Statement": [
  {
    "Sid": "AllowsAmazonQToGetSecret",
    "Effect": "Allow",
    "Action": [
      "secretsmanager:GetSecretValue"
    ],
    "Resource": [
      "arn:aws:secretsmanager:{region}:{{account_id}}:secret:{{secret_id}}"
    ]
  },
  {
    "Sid": "AllowsAmazonQToDecryptSecret",
    "Effect": "Allow",
    "Action": [
      "kms:Decrypt"
    ],
    "Resource": [
      "arn:aws:kms:{region}:{{account_id}}:key:{{key_id}}"
    ],
    "Condition": {
      "StringLike": {
        "kms:ViaService": [
          "secretsmanager.*.amazonaws.com"
        ]
      ]
    }
  },
  {
    "Sid": "AllowsAmazonQToIngestDocuments",
    "Effect": "Allow",
    "Action": [
      "qbusiness:BatchPutDocument",
      "qbusiness:BatchDeleteDocument"
    ],
    "Resource": "arn:aws:qbusiness:{region}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
  },
  {
    "Sid": "AllowsAmazonQToIngestPrincipalMapping",
    "Effect": "Allow",
    "Action": [
      "qbusiness:PutGroup",
      "qbusiness:CreateUser"
    ]
  }
]
"qbusiness:DeleteGroup",
"qbusiness:UpdateUser",
"qbusiness:ListGroups"
],
"Resource": [
"arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
"arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
"arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**
]
},
{
"Sid": "AllowsAmazonQToCreateAndDeleteNI",
"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]]"
]
},
{
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterface",
"ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringLike": {
"aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_{{*}}"
},
"ForAllValues:StringEquals": {
"aws:TagKeys": [
"AMAZON_Q"
]
}
}
},
{
"Sid": "QBusines3033319E3"
}
"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
  "ec2:CreateTags"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
  "StringEquals": {
    "ec2:CreateAction": "CreateNetworkInterface"
  }
}
},
{
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    }
  }
},
{
  "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeNetworkInterfaceAttribute",
    "ec2:DescribeVpcs",
    "ec2:DescribeRegions",
    "ec2:DescribeNetworkInterfacePermissions",
    "ec2:DescribeSubnets"
  ],
  "Resource": "***"
}
]

To allow Amazon Q to assume a role, you must also use the following trust policy:
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

### Connecting ServiceNow Online to Amazon Q

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 connect ServiceNow instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

**Learn more**

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.
Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

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 Q.
- Configured basic authentication credentials containing a username and password to allow Amazon Q to connect to your ServiceNow instance.
- Optional: Configured an OAuth 2.0 credential token that can identify Amazon Q using a username, password, and a generated client ID, and a client secret. For more information, see ServiceNow documentation on OAuth 2.0 authentication on the ServiceNow website.

In your AWS account, make sure you have:

- Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
- Stored your ServiceNow authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note

If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.
For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the ServiceNow Online page, enter the following information:

1. Name – Name your data source for easy tracking.

   **Note:** You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. In Source, enter the following information:
   - ServiceNow host – Enter your ServiceNow host name without the protocol. For example, example.service-now.com.
   - ServiceNow version – Select your ServiceNow version.

3. Authorization – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.

   **Note**
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).

4. Authentication – Choose between Basic authentication and OAuth 2.0 authentication and then enter the following information for your AWS Secrets Manager secret.
   - Secret name – A name for your secret.
   - 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 that you created in your ServiceNow account.
   - Choose Save and add secret.

5. Configure VPC and security group – optional – Choose whether you want to use a VPC. If you do, enter the following information:
a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

For more information, see [VPC](#).

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

For more information, see [IAM role](#).

8. **Sync scope** – Set the content that you want to sync.

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`, `kbnowledge_base=dfc19531bf2021003f07e2c1ac0739ab^text ISNOTEMPTY^EQ`, and `article_type=text^active=true^EQ`.

```
⚠️ Important

If you choose to crawl **Only public articles**, Amazon Q 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^EQ, 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, and priority=2^category=software^EQ.

  • **Include incidents based on short description filter** – Specify a regex pattern to include specific incidents.

  d. **In Additional configuration – optional:**

  • **ACL information** – Access control lists for entities that 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 that aren't selected. For public articles, ACL isn't applied.

  • **Attachment regex patterns** – Add regular expression patterns to include or exclude specific attached files of catalogs, knowledge articles, and incidents. You can add up to 100 patterns.
9. 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 Q for the first time, all content is synced by default.
   - **Full sync** – Sync all content regardless of the previous sync status.
   - **New, modified, or deleted content sync** – Only sync new, modified, and deleted content.

10. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

11. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

12. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:
   a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.
   b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

13. To finish connecting your data source to Amazon Q, select **Add data source**.

   You are taken to the **Data source details**, where you can view your data source configuration details.

14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

### Note

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.
Using the API

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**ServiceNow JSON schema**

The following is the ServiceNow JSON schema:

```json
{
  "$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",
              "dateFieldFormat"
            ]
          }
        }
      }
    }
  }
},

"required": [
  "repositoryEndpointMetadata"
]
"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",
 "dateFieldFormat"
]}}}}
}
"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": [  "indexFieldName",
"indexFieldType",
"dataSourceFieldName"
]  }
}
]
},
"required": [
  "fieldMappings"
]
},
"incident": {
  "type": "object",
  "properties": {
    "fieldMappings": [
      {
        "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"
      ]
    ]
  }
},
"ServiceNow Online": 867
"required": [
  "fieldMappings"
]
}
}
"additionalProperties": {
  "type": "object",
  "properties": {
    "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"
    }
  }
}
"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": [
    
  ]
}
```json
{  
  "pattern": "1.0.0",
  "required": [
    "connectionConfiguration",
    "repositoryConfigurations",
    "syncMode",
    "additionalProperties",
    "secretArn",
    "type"
  ]
}
```

The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>hostUrl</td>
<td>The ServiceNow host URL. For example, <code>your-domain.service-now.com</code>.</td>
</tr>
<tr>
<td>authType</td>
<td>The type of authentication you are using, either basicAuth or OAuth2.</td>
</tr>
<tr>
<td>servicenowInstanceVersion</td>
<td>The ServiceNow version you are using. You can choose between Tokyo, San Diego, Rome, and others.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• knowledgeArticle</td>
<td>A list of objects that map the attributes or field names of your ServiceNow knowledge articles, attachments, service catalog, and incidents to Amazon Q index field names.</td>
</tr>
<tr>
<td>• attachment</td>
<td></td>
</tr>
<tr>
<td>• serviceCatalog</td>
<td></td>
</tr>
<tr>
<td>• incident</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source. <strong>knowledgeArticleFilter</strong>, <strong>incidentQueryFilter</strong>, <strong>serviceCatalogQueryFilter</strong>, <strong>knowledgeArticleTitleRegExp</strong>, <strong>serviceCatalogTitleRegExp</strong>, <strong>incidentTitleRegExp</strong>, <strong>inclusionFileTypePatterns</strong>, <strong>exclusionFileTypePatterns</strong>, <strong>inclusionFileNamePatterns</strong>, <strong>exclusionFileNamePatterns</strong>, <strong>incidentStateType</strong>.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>isCrawlKnowledgeArticle</td>
<td>True to index ServiceNow knowledge articles, service catalogs, incidents, and attachments.</td>
</tr>
<tr>
<td>isCrawlKnowledgeArticleAttachment</td>
<td></td>
</tr>
<tr>
<td>includePublicArticlesOnly</td>
<td></td>
</tr>
<tr>
<td>isCrawlServiceCatalog</td>
<td></td>
</tr>
<tr>
<td>isCrawlServiceCatalogAttachment</td>
<td></td>
</tr>
<tr>
<td>isCrawlActiveServiceCatalog</td>
<td></td>
</tr>
<tr>
<td>isCrawlInactiveServiceCatalog</td>
<td></td>
</tr>
<tr>
<td>isCrawlActiveIncident</td>
<td></td>
</tr>
<tr>
<td>isCrawlInactiveIncident</td>
<td></td>
</tr>
<tr>
<td>applyACLForKnowledgeArticle</td>
<td></td>
</tr>
<tr>
<td>applyACLForServiceCatalog</td>
<td></td>
</tr>
<tr>
<td>applyACLForIncident</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>The type of data source. Specify SERVICENOWV2 as your data source type.</th>
</tr>
</thead>
<tbody>
<tr>
<td>enableIdentityCrawler</td>
<td>True to activate identity crawler. Identity crawler is activated by default. Crawling identity information on users and groups with access to specific documents is useful for user context filtering. Search results are filtered based on the user or their group access to documents. See <a href="#">Identity crawler</a> for more information.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| syncMode      | Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:  
  - Use `FORCED_FULL_CRAWL` to freshly re-crawl all content and replace existing content each time your data source syncs with your index  
  - Use `FULL_CRAWL` to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index |
| secretARN     | 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:  
  ```json  
  {  
    "username": "username",  
    "password": "password"  
  }  
  ```  
  If you use OAuth2 authentication, your secret must contain a JSON structure with the following keys:  
  ```json  
  {  
    "username": "username",  
    "password": "password",  
    "clientId": "client id",  
    "clientSecret": "client secret"  
  }  
  ``` |
| version       | The version of the template that's currently supported. |
ACL crawling

When you connect an ServiceNow data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your ServiceNow instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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're 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're mapped from the user emails as the IDs in ServiceNow.

For more information, see:

- Authorization
- Identity crawler
- Understanding User Store

IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the `BatchPutDocument` and `BatchDeleteDocument` operations to ingest documents.
- Permission to access the [User Store](#) API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    ...
  ]
}
```
Amazon Q

Developer Guide

{
"Sid": "AllowsAmazonQToGetSecret",
"Effect": "Allow",
"Action": [
"secretsmanager:GetSecretValue"
],
"Resource": [
"arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
]
},
{
"Sid": "AllowsAmazonQToDecryptSecret",
"Effect": "Allow",
"Action": [
"kms:Decrypt"
],
"Resource": [
"arn:aws:kms:{{region}}:{{account_id}}:key/[[key_id]]"
],
"Condition": {
"StringLike": {
"kms:ViaService": [
"secretsmanager.*.amazonaws.com"
]
}
}
},
{
"Sid": "AllowsAmazonQToIngestDocuments",
"Effect": "Allow",
"Action": [
"qbusiness:BatchPutDocument",
"qbusiness:BatchDeleteDocument"
],
"Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/
{{application_id}}/index/{{index_id}}"
},
{
"Sid": "AllowsAmazonQToIngestPrincipalMapping",
"Effect": "Allow",
"Action": [
"qbusiness:PutGroup",
"qbusiness:CreateUser",
"qbusiness:DeleteGroup",

ServiceNow Online

876


"qbusiness:UpdateUser",
"qbusiness:ListGroups"
],
"Resource": [
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**
]
},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "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]"
  ]
},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "AMAZON_Q"
      ]
    }
  }
},
{
  "Sid": "AllowsAmazonQToCreateTags",
To allow Amazon Q to assume a role, you must also use the following trust policy:
For more information on Amazon Q data source connector IAM roles, see [IAM roles for Amazon Q data source connectors](#).

**Connecting Slack to Amazon Q**

Slack is an enterprise communications app that lets users send messages and attachments through various public and private channels. You can connect your Slack instance to Amazon Q—using either the AWS Management Console, CLI, or the [CreateDataSource](#) API—and create an Amazon Q web experience.

**Learn more**

- For an overview of the Amazon Q web experience creation process, see [Configuring an application](#).
- For an overview of connector features, see [Data source connector concepts](#).
- For information about connector configuration best practices, see [Connector configuration best practices](#).
Topics

- Prerequisites
- Using the console
- Using the API
- ACL crawling
- IAM roles

Prerequisites

Before you begin, make sure that you have completed the following prerequisites.

In Slack, make sure you have:

- Created a Slack Bot User OAuth token or Slack User OAuth token. You can choose either token to connect Amazon Q to your Slack data source. 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.

- 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/ read permissions:

<table>
<thead>
<tr>
<th>User token scope</th>
<th>Bot token scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>channels:history</td>
<td>team:read</td>
</tr>
<tr>
<td>channels:read</td>
<td>channels:history</td>
</tr>
<tr>
<td>channels:write</td>
<td>groups:history</td>
</tr>
<tr>
<td>chat:write</td>
<td>im:history</td>
</tr>
<tr>
<td>emoji:read</td>
<td>mpim:history</td>
</tr>
<tr>
<td>files:read</td>
<td>chat:write</td>
</tr>
<tr>
<td>User token scope</td>
<td>Bot token scope</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>• files:write</td>
<td>• channels:manage</td>
</tr>
<tr>
<td>• groups:history</td>
<td>• groups:write</td>
</tr>
<tr>
<td>• groups:read</td>
<td>• groups:read</td>
</tr>
<tr>
<td>• groups:write</td>
<td>• im:write</td>
</tr>
<tr>
<td>• im:history</td>
<td>• im:read</td>
</tr>
<tr>
<td>• im:read</td>
<td>• files:write</td>
</tr>
<tr>
<td>• im:write</td>
<td>• files:read</td>
</tr>
<tr>
<td>• mpim:history</td>
<td>• users:read</td>
</tr>
<tr>
<td>• mpim:read</td>
<td>• links:read</td>
</tr>
<tr>
<td>• mpim:write</td>
<td>• channels:read</td>
</tr>
<tr>
<td>• reactions:write</td>
<td>• mpim:read</td>
</tr>
<tr>
<td>• team:read</td>
<td>• chat:write.customize</td>
</tr>
<tr>
<td>• usergroups:read</td>
<td>• channels:join</td>
</tr>
<tr>
<td>• usergroups:write</td>
<td>• emoji:read</td>
</tr>
<tr>
<td>• users.profile:read</td>
<td>• mpim:write</td>
</tr>
<tr>
<td>• users:read</td>
<td>• usergroups:read</td>
</tr>
<tr>
<td>• users:read.email</td>
<td>• usergroups:write</td>
</tr>
<tr>
<td>• auditlogs:read</td>
<td>• users.profile:read</td>
</tr>
</tbody>
</table>

**In your AWS account, make sure you have:**

- Created an [IAM role](#) for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.

- Stored your Slack authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.
Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.

For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Slack page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   *Note:* You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **In Source, Slack workspace team ID** – The team ID of your Slack workspace.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.

   *Note* Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).

4. **Authentication** – Enter the following information for your AWS Secrets Manager secret.

   a. **Secret name** – A name for your secret.

   b. For **Slack token** – Enter the authentication credential values you created in your Slack account.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.
b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

For more information, see [VPC](#).

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see [Identity crawler](#).

7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

For more information, see [IAM role](#).

8. In **Sync scope**, enter the following information:

   a. **Select type of content to crawl** – Select any combination of All channels, Public channels, Private channels, Group messages, and Private messages.

   b. **Select crawl start date** – Choose the date from which the Amazon Q connector will start crawling content.

   c. **Additional configuration – optional** – Configure the following settings:

      - In **Channels** (available only if you've chosen to crawl Channels), do the following:

        - **Channel ID/Name** – Choose between Channel ID and Channel Name.

        - **Note**
          
          You can choose to configure both.

        - For **Channel ID** – Enter the Channel ID. The Channel ID filter applies to both public and private channels.

        - For **Channel Name** – Choose the Channel type and enter the Channel name. You can select between Public channel and Private channel.

        - **Note**
          
          If you choose to configure filters for both Channel ID and Channel Name, the Amazon Q Slack connector will prioritize channel IDs over channel names.
If you choose to configure filters for either Channel ID or Channel Name, the Amazon Q Slack connector will ignore Private and Group messages even if you’ve chosen to crawl private and group messages in Sync scope.

- In Messages, for Select sync scope for content – Choose to Include bot messages, and/or Include archived messages.
- Regex patterns – Add regex patterns to include or exclude file names or file types. You can add a total of 100 patterns. Examples of regex patterns include:
  - File type – .pdf, .docx
  - File name – Hello*.txt, TestFile.*

9. 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 Q for the first time, all content is synced by default.
   - Full sync—Sync all content regardless of the previous sync status.
   - New, modified, or deleted content sync—Sync only new, modified, and deleted documents.

10. In Sync run schedule, for Frequency – Choose how often Amazon Q will sync with your data source. For more details, see Sync run schedule.

11. Tags - optional – Add tags to search and filter your resources or track your AWS costs. See Tags for more details.

12. Field mappings – A list of data source document attributes to map to your index fields. Add the fields from the Data source details page after you finish adding your data source. You can choose from two types of fields:
   a. Default – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can’t edit these.
   b. Custom – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see Field mappings.

13. To finish connecting your data source to Amazon Q, select Add data source.

   You are taken to the Data source details, where you can view your data source configuration details.
14. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.

**Note**

You can also choose to view CloudWatch logs for your data source sync job by selecting **View CloudWatch logs**. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

**Using the API**

You use the **CreateDataSource** action to connect a data source to your Amazon Q application. Then, you use the **configuration** parameter to provide a JSON schema with all other configuration information specific to your data source connector.

**Slack JSON schema**

The following is the Slack JSON schema:

```json
{
  "$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",
  "indexFieldFormat",
  "dataSourceFieldName"
],
"additionalProperties": {
  "type": "object",
}
"properties": {
  "isCrawlAcl": {
    "type": "boolean"
  },
  "fieldForUserId": {
    "type": "string"
  },
  "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": "^"\n    }
  ]
},
"lookBack": {
  "type": "string",
  "pattern": "^[0-9]+$"
},
"required": [
],
"syncMode": {
  "type": "string",
  "enum": [
    "FORCED_FULL_CRAWL",
    "FULL_CRAWL",
    "CHANGE_LOG"
  ]
},
"registration": {
  "required": [
  ],
  "anyOf": [
    {
      "type": "string",
      "pattern": "^[0-9]+$"
    },
    {
      "type": "string",
      "pattern": "^"\n    }
  ]
},
"channels": {
  "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$"
    },
    {
      "type": "string",
      "pattern": "^"\n    }
  ]
},
"lookBack": {
  "type": "string",
  "pattern": "^[0-9]+$"
},
"required": [
],
"syncMode": {
  "type": "string",
  "enum": [
    "FORCED_FULL_CRAWL",
    "FULL_CRAWL",
    "CHANGE_LOG"
  ]
},
"registration": {
  "required": [
  ],
  "anyOf": [
    {
      "type": "string",
      "pattern": "^[0-9]+$"
    },
    {
      "type": "string",
      "pattern": "^"\n    }
  ]
},
"channels": {
  "type": "array",
  "items": {
    "type": "string"
  }
},
"channelIdFilter": {
  "type": "array",
  "items": {
    "type": "string"
  }
},
"sinceDate": {
  "anyOf": [
    {
      "type": "string",
      "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}Z$"
    },
    {
      "type": "string",
      "pattern": "^"\n    }
  ]
},
"lookBack": {
  "type": "string",
  "pattern": "^[0-9]+$"
},
"required": [
],
"syncMode": {
  "type": "string",
  "enum": [
    "FORCED_FULL_CRAWL",
    "FULL_CRAWL",
    "CHANGE_LOG"
  ]
},
"registration": {
  "required": [
  ],
  "anyOf": [
    {
      "type": "string",
      "pattern": "^[0-9]+$"
    },
    {
      "type": "string",
      "pattern": "^"\n    }
  ]
},
"channels": {
  "type": "array",
  "items": {
    "type": "string"
  }
}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint for the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>teamId</td>
<td>The Slack team ID you copied from your Slack main page URL.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• All</td>
<td>A list of objects that map the attributes or field names of your Slack pages and assets to Amazon Q index field names.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>isCrawlAcl</td>
<td>Specify true to crawl access control information from documents.</td>
</tr>
<tr>
<td>fieldForUserId</td>
<td>Specify field to use for UserId for ACL crawling.</td>
</tr>
<tr>
<td>• inclusionPatterns</td>
<td>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.</td>
</tr>
<tr>
<td>• exclusionPatterns</td>
<td>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 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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>crawlBotMessages</td>
<td>true to crawl Slack bot messages.</td>
</tr>
<tr>
<td>excludeArchived</td>
<td>true to exclude archived messages from crawl.</td>
</tr>
<tr>
<td>channelFilter</td>
<td>The type of channel that you want to index whether <code>private_channel</code> or <code>public_channel</code>.</td>
</tr>
<tr>
<td>conversationType</td>
<td>The type of channel that you want to index whether <code>PUBLIC_CHANNEL</code>, <code>PRIVATE_CHANNEL</code>, <code>GROUP_MESSAGE</code> and <code>DIRECT_MESSAGE</code>.</td>
</tr>
<tr>
<td>sinceDate</td>
<td>You can choose to configure a <code>sinceDate</code> parameter so that the Slack connector crawls content based on a specific <code>sinceDate</code>.</td>
</tr>
<tr>
<td>lookBack</td>
<td>You can choose to configure a <code>lookBack</code> parameter so that the Slack connector crawls <code>lookBack</code> content.</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>syncMode</td>
<td>Specify whether Amazon Q should update your index by syncing all documents or only new, modified, and deleted documents. You can choose between the following options:</td>
</tr>
<tr>
<td></td>
<td>- Use FORCED_FULL_CRAWL to freshly re-crawl all content and replace existing content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>- Use FULL_CRAWL to incrementally crawl only new, modified, and deleted content each time your data source syncs with your index.</td>
</tr>
<tr>
<td></td>
<td>- Use CHANGE_LOG to incrementally crawl only new and modified content each time your data source syncs with your index.</td>
</tr>
<tr>
<td>type</td>
<td>The type of data source. Specify SLACK as your data source type.</td>
</tr>
<tr>
<td>enableIdentityCrawler</td>
<td>Specify true to use the Amazon Q identity crawler to sync identity/principal information on users and groups with access to specific documents.</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 Slack. The secret must contain a JSON structure with the following keys:</td>
</tr>
<tr>
<td></td>
<td>```</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;slackToken&quot;: &quot;token&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>```</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>version</td>
<td>The version of this template that's currently supported.</td>
</tr>
</tbody>
</table>

**ACL crawling**

When you connect an Slack data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Slack instance. If you choose to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

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.

For more information, see:

- [Authorization](#)
- [Identity crawler](#)
- [Understanding User Store](#)

**IAM roles**

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

- Permission to access the BatchPutDocument and BatchDeleteDocument operations to ingest documents.
- Permission to access the User Store API operations to ingest user and group access control information from documents.
- Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
- **(Optional)** If you're using Amazon VPC, permission to access your Amazon VPC.
{  
  "Version": "2012-10-17",
  "Statement": [

    {  
      "Sid": "AllowsAmazonQToGetSecret",
      "Effect": "Allow",
      "Action": [  
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [  
        "arn:aws:secretsmanager:{{region}}:{{account_id}}:secret:[[secret_id]]"
      ]
    },

    {  
      "Sid": "AllowsAmazonQToDecryptSecret",
      "Effect": "Allow",
      "Action": [  
        "kms:Decrypt"
      ],
      "Resource": [  
        "arn:aws:kms:{{region}}:{{account_id}}:key/[[key_id]]"
      ],
      "Condition": {  
        "StringLike": {  
          "kms:ViaService": [  
            "secretsmanager.*.amazonaws.com"
          ]
        }
      }
    },

    {  
      "Sid": "AllowsAmazonQToIngestDocuments",
      "Effect": "Allow",
      "Action": [  
        "qbusiness:BatchPutDocument",
        "qbusiness:BatchDeleteDocument"
      ],
      "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
    },

    {  
      "Sid": "AllowsAmazonQToIngestPrincipalMapping",
      "Effect": "Allow",
      "Action": [  
        "qbusiness:PutPrincipalMapping"
      ],
      "Resource": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}/index/{{index_id}}"
    }
  ]
}
"Action": [
  "qbusiness:PutGroup",
  "qbusiness:CreateUser",
  "qbusiness:DeleteGroup",
  "qbusiness:UpdateUser",
  "qbusiness:ListGroup"
],
"Resource": [
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}",
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
  "arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**
],
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNI",
  "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]]"
  
},
{
  "Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface"
],
  "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
  "Condition": {
    "StringLike": {
      "aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
    },
    "ForAllValues:StringEquals": {
      "aws:TagKeys": [
        "AMAZON_Q"
      ]
    }
  }
}
{
    "Sid": "AllowsAmazonQToCreateTags",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "CreateNetworkInterface"
        }
    }
},
{
    "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateNetworkInterfacePermission"
    ],
    "Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
    "Condition": {
        "StringLike": {
            "aws:ResourceTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
        }
    }
},
{
    "Sid": "AllowsAmazonQToDescribeResourcesForVPC",
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:DescribeVpcs",
        "ec2:DescribeRegions",
        "ec2:DescribeNetworkInterfacePermissions",
        "ec2:DescribeSubnets"
    ],
    "Resource": "*"
}
To allow Amazon Q to assume a role, you must also use the following trust policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "qbusiness.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{source_account}}"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:qbusiness:{{region}}:{{source_account}}:application/{{application_id}}"
        }
      }
    }
  ]
}
```

For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

**Connecting Zendesk to Amazon Q**

Zendesk is a customer relationship management system that helps businesses automate and enhance customer support interactions. You can connect Zendesk instance to Amazon Q—using either the AWS Management Console or the CreateDataSource API—and create an Amazon Q web experience.

**Learn more**

- For an overview of the Amazon Q web experience creation process, see Configuring an application.
• For an overview of connector features, see Data source connector concepts.
• For information about connector configuration best practices, see Connector configuration best practices.

Topics
• Prerequisites
• Using the console
• Using the API
• ACL crawling
• IAM roles

Prerequisites
Before you begin, make sure that you have completed the following prerequisites.

In Zendesk, make sure you have:

• Created a Zendesk Suite (Professional/Enterprise) administrative account.
• Copied your Zendesk host URL. For example, https://{sub-domain}{host}.zendesk.com/. You need this URL to allow Amazon Q to connect with your Zendesk data source.
• Generated Zendesk OAuth 2.0 credentials containing a client id, client secret, username, and password. You need these credentials to authenticate Amazon Q to access Zendesk.

In your AWS account, make sure you have:

• Created an IAM role for your data source and, if using the Amazon Q API, noted the ARN of the IAM role.
• Stored your Zendesk authentication credentials in an AWS Secrets Manager secret and, if using the Amazon Q API, noted the ARN of the secret.

Note
If you’re a console user, you can create the IAM role and Secrets Manager secret as part of configuring your Amazon Q application on the console.
For a list of things to consider while configuring your data source, see Data source connector configuration best practices.

Using the console

On the Zendesk page, enter the following information:

1. **Name** – Name your data source for easy tracking.
   
   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.

2. **Source** – Enter your Zendesk URL. For example, https://{sub-domain}({host}).zendesk.com/.

3. **Authorization** – Choose whether Amazon Q will crawl user and group access control list (ACL) information from your data source. Amazon Q can use this information to only generate responses from documents your end users have access to. See Authorization for more details.

   Note
   
   Using ACL data to filter responses is not a replacement for user authentication and authorization for your application. For information on setting up identity management for Amazon Q, see Integrating with an Identity Provider (IdP).

4. **Authentication** – Enter a name for your secret, a client ID, client secret, username, and password.

5. **Configure VPC and security group – optional** – Choose whether you want to use a VPC. If you do, enter the following information:

   a. **Subnets** – Select up to 6 repository subnets that define the subnets and IP ranges the repository instance uses in the selected VPC.

   b. **VPC security groups** – Choose up to 10 security groups that allow access to your data source. Ensure that the security group allows incoming traffic from Amazon EC2 instances and devices outside your VPC. For databases, security group instances are required.

   For more information, see VPC.

6. **Identity crawler** – Choose to activate Amazon Q identity crawler to sync identity information. For more information, see Identity crawler.
7. **IAM role** – Choose an existing IAM role or create an IAM role to access your repository credentials and index content.

   For more information, see [IAM role](#).

8. **Sync scope** – Set the content that you want to sync.

   - **Additional configuration – optional** – Configure the following settings:
     - **Change log** – Select to update your index instead of syncing all your files.
     - **Organization name** – Enter the Zendesk organization names to filter your sync.
     - **Sync start date** – The date from which you want to index your content.
     - **Regex patterns** – Regular expression patterns to include or exclude certain files. You can add up to 100 patterns.

9. In **Sync run schedule**, for **Frequency** – Choose how often Amazon Q will sync with your data source. For more details, see [Sync run schedule](#).

10. **Tags - optional** – Add tags to search and filter your resources or track your AWS costs. See [Tags](#) for more details.

11. **Field mappings** – A list of data source document attributes to map to your index fields. Add the fields from the **Data source details** page after you finish adding your data source. You can choose from two types of fields:

    a. **Default** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can't edit these.
    
    b. **Custom** – Automatically created by Amazon Q on your behalf based on common fields in your data source. You can edit these. You can also create and add new custom fields.

   For more information, see [Field mappings](#).

12. To finish connecting your data source to Amazon Q, select **Add data source**.

    You are taken to the **Data source details**, where you can view your data source configuration details.

13. In **Data source details**, choose **Sync now** to allow Amazon Q to begin syncing (crawling and ingesting) data from your data source. When the sync job finishes, your data source is ready to use.
## Note

You can also choose to view CloudWatch logs for your data source sync job by selecting View CloudWatch logs. If you get a Resource not found exception when you try to view your CloudWatch logs for a data source sync job in progress, it can be because the CloudWatch logs are not available yet. Wait for some time and check again.

## Using the API

You use the [CreateDataSource](#) action to connect a data source to your Amazon Q application.

Then, you use the configuration parameter to provide a JSON schema with all other configuration information specific to your data source connector.

### JSON schema

The following is the Zendesk JSON schema:

```json
{
    "$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"
            ]
        }
    }
}
```

Zendesk
"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"
                  }
                }
              }
            ]
          }
        }
      }
    }
  }
}
"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",
              "dateFieldFormat"
            ]
          }
        ]
      }
    }
  }
},
"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": ["object"]
         }
      }
   }
}
"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"
]
},
"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"
                     }
                  }
               }
            ]
         }
      }
   }
}
"required": [
  "indexFieldName",
  "indexFieldType",
  "dataSourceFieldName"
],
"dateFieldFormat": {
  "type": "string",
  "pattern": "dd-MM-yyyy HH:mm:ss"
},
"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"
              }
            }
          }
        ]
      }
    }
  }
}

"dataSourceFieldName"
]
]
]
,
"required": [
 "fieldMappings"
]
]

"secretArn": {
 "type": "string",
 "minLength": 20,
 "maxLength": 2048
},
"additionalProperties": {
 "type": "object",
 "properties": {
 "isCrawlAcl": {
 "type": "boolean"
 },
 "fieldForUserId": {
 "type": "string"
 },
 "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"]
},
"enableIdentityCrawler": {
  "type": "boolean"
},
"version": {
  "type": "string",
  "anyOf": [
    {
      "pattern": "1.0.0"
    }
  ]
}
The following table provides information about important JSON keys to configure.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionConfiguration</td>
<td>Configuration information for the endpoint of the data source.</td>
</tr>
<tr>
<td>repositoryEndpointMetadata</td>
<td>The endpoint information for the data source.</td>
</tr>
<tr>
<td>hostURL</td>
<td>The Zendesk host URL. For example, <a href="https://yoursubdomain.zendesk.com">https://yoursubdomain.zendesk.com</a>.</td>
</tr>
<tr>
<td>repositoryConfigurations</td>
<td>Configuration information for the content of the data source. For example, configuring specific types of content and field mappings.</td>
</tr>
<tr>
<td>• ticket</td>
<td>A list of objects that map attributes or field names of Zendesk tickets to Amazon Q index field names. The Zendesk data source field names must exist in your Zendesk custom metadata.</td>
</tr>
<tr>
<td>• ticketComment</td>
<td></td>
</tr>
<tr>
<td>• ticketCommentAttachment</td>
<td></td>
</tr>
<tr>
<td>• article</td>
<td></td>
</tr>
<tr>
<td>• articleComment</td>
<td></td>
</tr>
<tr>
<td>• articleAttachment</td>
<td></td>
</tr>
<tr>
<td>• communityTopic</td>
<td></td>
</tr>
<tr>
<td>• communityPost</td>
<td></td>
</tr>
<tr>
<td>• communityPostComment</td>
<td></td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------</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 Zendesk. The secret must contain a JSON structure with the following keys: host URL, client ID, client secret, username, and password.</td>
</tr>
<tr>
<td>additionalProperties</td>
<td>Additional configuration options for your content in your data source.</td>
</tr>
<tr>
<td>organizationFilter</td>
<td>If you want, you can choose to index tickets that exist within a specific <strong>Organization</strong>.</td>
</tr>
<tr>
<td>sinceDate</td>
<td>If you want, you can configure a sinceDate parameter so that the Zendesk connector will crawl based on the sinceDate.</td>
</tr>
<tr>
<td>inclusionPatterns</td>
<td>A list of regular expression patterns to <em>include</em> specific 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.</td>
</tr>
<tr>
<td>exclusionPatterns</td>
<td>A list of regular expression patterns to <em>exclude</em> specific 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.</td>
</tr>
</tbody>
</table>
### Configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>• isCrawlTicket</td>
<td>Input TRUE to index.</td>
</tr>
<tr>
<td>• isCrawlTicketComment</td>
<td></td>
</tr>
<tr>
<td>• isCrawlTicketCommentAttachment</td>
<td></td>
</tr>
<tr>
<td>• isCrawlArticle</td>
<td></td>
</tr>
<tr>
<td>• isCrawlArticleComment</td>
<td></td>
</tr>
<tr>
<td>• isCrawlArticleAttachment</td>
<td></td>
</tr>
<tr>
<td>• isCrawlCommunityTopic</td>
<td></td>
</tr>
<tr>
<td>• isCrawlCommunityPost</td>
<td></td>
</tr>
<tr>
<td>• isCrawlCommunityPostComment</td>
<td></td>
</tr>
<tr>
<td><strong>type</strong></td>
<td>Specify ZENDESK as your data source type.</td>
</tr>
<tr>
<td><strong>useChangeLog</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>enableIdentityCrawler</strong></td>
<td>true to activate identity crawler. Identity crawler is activated by default. Crawling identity information on users and groups with access to certain documents is useful for user context filtering. Search results are filtered based on the user or their group access to documents. See <a href="#">Identity crawler</a> for more information.</td>
</tr>
</tbody>
</table>

### ACL crawling

When you connect an Zendesk data source to Amazon Q, Amazon Q crawls ACL information attached to a document (user and group information) from your Zendesk instance. If you choose
to activate ACL crawling, the information can be used to filter chat responses to your end user's document access level.

The group and user IDs are mapped as follows:

• \texttt{\_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.
• \texttt{\_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.

For more information, see:

• Authorization
• Identity crawler
• Understanding User Store

IAM roles

To connect your data source connector to Amazon Q, you must give Amazon Q an IAM role that has the following permissions:

• Permission to access the \texttt{BatchPutDocument} and \texttt{BatchDeleteDocument} operations to ingest documents.
• Permission to access the User Store API operations to ingest user and group access control information from documents.
• Permission to access your AWS Secrets Manager secret to authenticate your data source connector instance.
• \textbf{(Optional)} If you're using Amazon VPC, permission to access your Amazon VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQToGetSecret",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ]
    }
  ]
}
```
"Resource": [
  "arn:aws:secretsmanager:{region}:{account_id}:secret:[[secret_id]]"
],


"arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}",
"arn:aws:qbusiness:{{region}}:{{account_id}}:application/{{application_id}}/index/{{index_id}}/data-source/**"
]}
{
"Sid": "AllowsAmazonQToCreateAndDeleteNI",
"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]]"
]
},
{
"Sid": "AllowsAmazonQToCreateAndDeleteNIForSpecificTag",
"Effect": "Allow",
"Action": [
"ec2:CreateNetworkInterface",
"ec2:DeleteNetworkInterface"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",
"Condition": {
"StringLike": {
"aws:RequestTag/AMAZON_Q": "qbusiness_{{account_id}}_{{application_id}}_*"
},
"ForAllValues:StringEquals": {
"aws:TagKeys": [
"AMAZON_Q"
]
}
}
},
{
"Sid": "AllowsAmazonQToCreateTags",
"Effect": "Allow",
"Action": [
"ec2:CreateTags"
],
"Resource": "arn:aws:ec2:{{region}}:{{account_id}}:network-interface/**",}
"Condition": {  
  "StringEquals": {  
    "ec2:CreateAction": "CreateNetworkInterface"  
  }  
},  
{  
  "Sid": "AllowsAmazonQToCreateNetworkInterfacePermission",  
  "Effect": "Allow",  
  "Action": [  
    "ec2:CreateNetworkInterfacePermission"  
  ],  
  "Resource": "arn:aws:ec2:{region}:{{account_id}}:network-interface/*",  
  "Condition": {  
    "StringLike": {  
      "aws:ResourceTag/AMAZON_Q": "qbusiness_{account_id}_{application_id}_*"  
    }  
  }  
},  
{  
  "Sid": "AllowsAmazonQToDescribeResourcesForVPC",  
  "Effect": "Allow",  
  "Action": [  
    "ec2:DescribeNetworkInterfaces",  
    "ec2:DescribeAvailabilityZones",  
    "ec2:DescribeNetworkInterfaceAttribute",  
    "ec2:DescribeVpcs",  
    "ec2:DescribeRegions",  
    "ec2:DescribeNetworkInterfacePermissions",  
    "ec2:DescribeSubnets"  
  ],  
  "Resource": "*"  
}  
}

To allow Amazon Q to assume a role, you must also use the following trust policy:

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsAmazonQServicePrincipal",
    }
  ]
}
For more information on Amazon Q data source connector IAM roles, see IAM roles for Amazon Q data source connectors.

Understanding Amazon Q User Store

With the Amazon Q User Store feature, end users see Amazon Q chat responses generated only from the documents that they have access to within an Amazon Q application. To achieve this, Amazon Q creates a mapping within the data sources attached to that application. The mapping is between every unique user accessing the application and all the user IDs and user groups that they are associated with. Amazon Q stores this principal mapping information in its internal User Store. During chat, Amazon Q uses the mapping information to return answers that are scoped to a user’s identity.

When you use the API, you use the User Store API actions to customize and configure your user management solution. For more details, see Using User Store APIs.

When you use the console, you can configure Amazon Q to automatically crawl user and group information using Authorization and Identity crawling features during the connector setup process. You can’t create, add, or customize users and groups to the user store using the AWS Management Console.
Note

The User Store feature is not available for the Amazon S3 and Amazon Q Web Crawler connectors that are used with Amazon Q. For more information about using access control information for user identity specific chat responses for these connectors, see Amazon S3 and Amazon Q Web Crawler.

Topics

• Principal mapping
• How the User Store works

Principal mapping

Amazon Q uses principal mapping to map users and groups with permissions to access an Amazon Q application to their user ids and group membership information within the data sources that are connected to the application.

Although user and group mapping is a synchronous, simultaneous process, the following sections explain them separately for conceptual clarity.

Topics

• User mapping
• Group mapping

User mapping

Each Amazon Q application can have multiple data sources connected to it. Each data source can have specific users and groups configured within it. Additionally, a user can be associated with multiple groups within a data source, or be attached to multiple groups across multiple data sources. A user attached to multiple data sources can also have different user IDs within these data sources.

A unique end user who signs in to an Amazon Q application must see only chat responses generated from documents that they have access to. To achieve that objective, Amazon Q maps their user IDs and group IDs within each data source to their identity provider (IdP) login
credentials. Then, Amazon Q creates a universally unique identifier (UUID) to assign to each user. Using the UUID that it creates, Amazon Q stores a comprehensive mapping of the user's group membership in an application. During chat, Amazon Q checks this UUID that's stored in its user store and retrieves user access information to generate chat responses.

The User Store feature also supports the following user management scenarios:

- **An end user leaves your organization.**

  When an end user leaves your organization, you can choose to delete the user from your user store.

- **An end user leaves your organization, and their email gets recycled.**

  Because User Store assigns each user a UUID for secure and accurate chat responses, email recycling doesn't impact the content that a user sees. Any new user within your application that's using a recycled email ID will be assigned a new UUID to be used for response generation.

- **An end user with multiple login IDs needs chat content generated from documents they access using both these login IDs.**

  With User Store, you can store user aliases attached to end user UUIDs. For example, a username Saanvi Sarkar uses two login IDs to sign in to Amazon Q—`saanvi_sarkar` and `saanvi_s`. You can store both IDs under the same UUID to ensure their chat responses are generated from content that they access using both login IDs.

**Group mapping**

Each Amazon Q application can have multiple data sources attached to it. Each data source in an Amazon Q application can have multiple groups attached to it. Multiple groups can repeat across multiple data sources. Additionally, each group across data sources can also contain multiple subgroups. Each Amazon Q application also has an associated identity provider (IdP) that can contain group information for the users accessing the application.

A unique end user signing in to an Amazon Q application must see only chat responses generated from documents within groups that they have access to. To achieve that objective, Amazon Q does the following:

- Automatically crawls local groups and their associated relationships from data sources during the connector configuration process.
• Provides you with API operations to map your end users group and subgroup membership details within each data source to their IdP group membership.

Then, Amazon Q creates a unique user identifier (UUID) to assign to each user. Under the UUID, Amazon Q stores a comprehensive mapping of the user’s group membership in an application. During chat, Amazon Q checks this UUID that's stored in its user store and quickly retrieves group access information to generate chat responses.

The User Store feature supports the following group management scenarios:

• **Your users mapped to all groups that they have access to within an Amazon Q application.**

  Amazon Q crawls all groups that a user has access to in a data source and stores this information under a user's UUID.

• **Create a subgroup of users within your application.**

  For example, for a group called company_employees, you might want to create a subgroup summer_interns and specify group level access for the subgroup. You might also want to group your interns into further subgroups like product_interns and engineering_interns.

• **Map your data source groups to your IdP groups.**

  A unique end user signing in to an Amazon Q application must see only chat responses generated from documents within groups they have access to. To support that objective, you can use Amazon Q to map your end users group membership details within each data source to their IdP group membership.

**Note**

Amazon Q doesn't interact or crawl this information from your IdP automatically. To ingest the relationship between data source groups and IdP groups, use the Amazon Q API.

### How the User Store works

The following overview describes how principal mapping works by using either the console or the Amazon Q API.
Using the console

Each document has access control list (ACL) information (which users and groups have access to it) attached to it as metadata. When you configure an Amazon Q data source connector, you can crawl this ACL information by activating the Authorization toggle.

When you crawl this ACL information, Amazon Q stores it in its internal user store to assess which user ids have access to a document. If you choose to not crawl ACL information, all documents are considered public.

Each data source also contains information about the users and groups which have access to it. During data source connector configuration, you can choose to crawl the information about users and groups attached to each data source by using the Identity crawler feature.

Amazon Q stores this crawled identity information in the user store and uses it to match and map user and group ids. You can only use the Identity crawler feature if you also crawl ACLs using the Authorization feature.

If you use the console, you must re-sync your data to your index to capture any changes in the ACL and user and group membership within your data source.

Using the API

When you configure your Amazon Q application, you use the following API operations to create your principal mapping solution:

User management

- CreateUser – Creates a universally unique identifier (UUID) that's mapped to a list of local user IDs within a data source.
- DeleteUser — Deletes a UUID that's mapped to a user.
- UpdateUser – Updates local user IDs within a data source that are mapped to a UUID.
- GetUser – Lists information associated with a user ID.

Group management
• **PutGroup** – Creates, or updates, a mapping of users to groups, or groups to subgroups. You can use this API operation to:
  • Map a group from groups in the data source to groups in your IdP.
  • Map a list of users and sub groups (for example, *Interns*) to a group (for example, *Interns 2023*).

• **DeleteGroup** – Deletes a group or a subgroup.

• **GetGroup** – Lists information about a group.

### Using Amazon VPC with Amazon Q connectors

Amazon Q 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 Q creates an elastic network interface that it uses to securely communicate with your data source within your VPC.

To set up an Amazon Q 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 Q 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 Q data source connectors support Amazon VPC.

**Important**

Amazon Q currently doesn't support AWS PrivateLink.

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 Q 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 Q connectors
- Set up an Amazon Q data source to connect to Amazon VPC
- Using Amazon VPC with an Amazon S3 data source
- Connecting to a database in a VPC
- Troubleshooting VPC connection issues

Configuring Amazon VPC support for Amazon Q connectors

To configure Amazon VPC for use with your Amazon Q connectors, take the following steps.

Steps

- Step 1. Create Amazon VPC subnets for Amazon Q
- Step 2. Create Amazon VPC security groups for Amazon Q
- Step 3. Configure your external data source and Amazon VPC

Step 1. Create Amazon VPC subnets for Amazon Q

Create or choose an existing Amazon VPC subnet that Amazon Q 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

Your data source must be accessible from the subnets that you provided to Amazon Q 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 Q 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 Q 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 Q

To connect your Amazon Q data source connector to Amazon VPC, you must prepare one or more security groups from your VPC to assign to Amazon Q. The security groups will be associated to the elastic network interface created by Amazon Q. This network interface controls inbound and outbound traffic to and from Amazon Q when accessing the Amazon VPC subnets.

Make sure that your security group's outbound rules allow the traffic from Amazon Q 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 Q 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 Q 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 Q 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 Q 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 Q. 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
Set up an Amazon Q data source to connect to Amazon VPC

When you add a new data source in Amazon Q, you can use the Amazon VPC feature if the selected data source connector supports this feature.

You can set up a new Amazon Q data source with Amazon VPC enabled by using the AWS Management Console or the Amazon Q 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/](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 Q 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 data sources.

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 Q data source connector to work properly.

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 Q 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 Q 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
more information about configuring an Amazon Q Amazon S3 connector with Amazon VPC, see Using Amazon VPC with Amazon S3.

For Amazon Q 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 Q. 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
- (Optional) Step 2: Configure Amazon S3 bucket policy
- Step 3: Create a test Amazon S3 data source connector

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 Q 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 Q 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 qbusiness-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 Q 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 `qbusiness-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 Q 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 Q 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 Q.

Amazon Q 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 Q.
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.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::bucket-name/*",
            "Condition": {
                "StringNotEquals": {
                    "aws:SourceVpce": "vpce-id"
                }
            }
        }
    ]
}
```
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)** – qbusiness-s3-example-vpc
- **Subnets** – qbusiness-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.

**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 Q 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 Q 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.

For instructions on how to create a data source for a MySQL database, see MySQL.

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 Q. 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](#)
- [Step 2: Connect to Amazon EC2 instance](#)
- [Step 3: Test Amazon S3 access](#)

**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/](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 Q.
   b. For **Firewall (security groups)**, choose **Select existing security group**. Then, select the security group that you assigned to Amazon Q.
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 your VPC 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.
Enhancing an Amazon Q application

After you finish configuring your application, you can optionally choose to enhance it.

You can choose from the following available enhancements:

- **Document enrichment** – Control document attribute ingestion and build customized data solutions.
- **Guardrails** – Customize blocked topics and choose the knowledge sources your web experience uses for responses.
- **Plugins** – Enable your end users to perform specific tasks related to third-party services from within their web experience chat—like creating Jira tickets.

**Topics**

- [Document enrichment in Amazon Q](#)
- [Configuring plugins with Amazon Q](#)
- [Admin controls and guardrails in Amazon Q](#)

**Document enrichment in Amazon Q**

The Amazon Q *document enrichment* feature helps you control both what documents and document attributes are ingested into your index and also how they're ingested. Using document enrichment, you can create, modify, or delete document attributes and document content when you ingest them into your Amazon Q index.

Document enrichment offers two kinds of methods that you can use for your solution:

- **Configure basic operations** – Use basic operations to add, update, or delete document attributes from your data. For example, you can scrub personally identifiable information (PII) by choosing to delete any document attributes related to PII.
- **Configure Lambda functions** – Use a preconfigured Lambda function to perform more customized, advanced document attribute manipulation logic to your data. For example, your enterprise data might be stored as scanned images. In that case, you can use a Lambda function to run Optical Character recognition (OCR) on the scanned documents to extract text from them. Then, each scanned document is treated as a text document during ingestion. Finally, during
Amazon Q will factor the textual data extracted from the scanned documents when it generates responses.

When you implement your solution, you can choose to use both document enrichment methods together. That is, you can use basic operations to do a first parse of your data and then use a Lambda function for more complex operations. For example, you could first use a basic function to remove all PII information from your documents using document attributes. Then, use a Lambda function to extract text from scanned documents.

Document enrichment is supported both on the AWS Management Console and by Amazon Q API actions. If you use the console, you can only enrich documents connected to your application using an Amazon Q data source.

Topics

- How document enrichment works
- Using basic operations for document enrichment
- Using Lambda functions

How document enrichment works

To understand and use document enrichments, you should be familiar with the key Amazon Q concepts that this topic outlines.

Topics

- Document enrichment concepts
- Document enrichment process overview

Document enrichment concepts

Amazon Q extracts document attributes from any document that you ingest into an Amazon Q index. Document attributes or structural metadata can include document title, document type, and time and date created. You can map document attributes to fields in an Amazon Q index to better structure your data for retrieval and chat. For more information, see Document attributes and types and Filtering using document attributes.
Note
Although document attributes and index fields are distinct concepts, in practice they’re used interchangeably because their values overlap and they structurally correspond to each other. That is, document attributes == document metadata == index fields.

Document enrichment process overview

The overall process of document enrichment is as follows:

• You configure document enrichment when you create or update your Amazon Q data source, or add or upload your documents directly into Amazon Q index. The exact process for configuration depends on the methods you choose:
  • If you use the API and want to configure document enrichment for a data source connector, you use the CreateDataSource and UpdateDataSource operations to provide your configuration details.
  • If you use the API and choose to directly upload documents into your index using the BatchPutDocument operation, you must configure document enrichment with each request.
  • If you use the console, can only configure document enrichment for a data source connected to your Amazon Q application. You select Document enrichments under Enhancements from the left navigation pane and configure enrichments. You can choose to use both configuration options or either one. You can also choose whether you want to apply your configuration to the original pre-extraction data or to the structured post-extraction data.
• After you configure and activate your document enrichment configuration, you can use inline configuration or basic logic to alter your data. For more information, see Using basic operations.
• If you chose to configure advanced data manipulation by using a Lambda function, Amazon Q applies the configured function (depending on what you’ve chosen) to either your original pre-extraction data or your structured post-extraction data. For more information, see Using Lambda functions.
• Finally, your altered and enriched documents are ingested into your Amazon Q index.

If a configuration isn’t valid during any point in this process, Amazon Q returns an error.
Using basic operations for document enrichment

With document enrichment, you can use basic operations to manipulate document attributes. For example, you can remove document attribute values, modify attribute values using conditions, or create document attributes.

Note

Amazon Q can’t create a target document attribute field if it isn’t already created as an index field.

Topics

- Basic operations using the Amazon Q API
- Basic operations using the Amazon Q console
- Examples of basic operations
- Code examples of basic operations

Basic operations using the Amazon Q API

To apply basic logic, you specify your document attribute configuration using the `DocumentAttributeTarget` object when you use either the `BatchPutDocument` API operation or the `CreateDataSource` operation. Use the following parameters to create your configuration:

- `key` – The target field that you want to manipulate. For example, the key `Department` is a field or attribute that holds all the department names associated with the documents.
- `value` – The target value for your target attribute.
- `attributeValueOperator` – To delete an existing target value, set to `DELETE`. The default value for this parameter is `UPDATE`.

If a specific condition is met, you can also specify a value to use in the target field. Set the condition using the `DocumentAttributeCondition` object. For example, if the `Source_URI` field contains `financial` in its URI value, you can choose to prefill the target field `Department` with the target value `Finance` for the document.

For more information, see the following topics in the `Amazon Q API Reference`:
• BatchPutDocument
• CreateDataSource
• DocumentAttributeTarget
• DocumentAttributeCondition

Basic operations using the Amazon Q console

To apply basic logic using the console

1. Sign in to the AWS Management Console and open the Amazon Q console.
2. In Applications, select the name of your application from the list of applications.
3. From the left navigation menu, choose Enhancements, and then choose Document enrichments.
5. In Configure basic operations, for Document enrichment source, choose a data source connected to your application.
6. To apply basic manipulations to your document fields and content, go to Configure basic operations.
7. Choose Next to save your configuration.

Examples of basic operations

This section provides two examples of basic operations.

Example 1: Removing customer identification numbers associated with the documents

The following is an example of using a basic operation to remove all customer identification numbers in the document field called Customer_ID.

The following table shows the data before basic manipulation is applied.

<table>
<thead>
<tr>
<th>Document_ID</th>
<th>Body_Text</th>
<th>Customer_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Example text</td>
<td>CID1234</td>
</tr>
<tr>
<td>2</td>
<td>Example text</td>
<td>CID1235</td>
</tr>
</tbody>
</table>
### Example 2: Creating and prefilling the Department field with department names associated with the documents using a condition

The following is an example of using basic logic to create a field called Department and prefilling the field with the department names based on information from the Source_URI field. This example uses the condition that, if the Source_URI field contains financial in its URI value, then the target field Department is prefilled with the target value Finance for the document.

The following table shows the data before basic manipulation is applied.

<table>
<thead>
<tr>
<th>Document_ID</th>
<th>Body_Text</th>
<th>Source_URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Example text</td>
<td>financial/1</td>
</tr>
<tr>
<td>2</td>
<td>Example text</td>
<td>financial/2</td>
</tr>
<tr>
<td>3</td>
<td>Example text</td>
<td>financial/3</td>
</tr>
</tbody>
</table>

The following table shows the data after basic manipulation is applied.

<table>
<thead>
<tr>
<th>Document_ID</th>
<th>Body_Text</th>
<th>Source_URI</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Example text</td>
<td>financial/1</td>
<td>Finance</td>
</tr>
</tbody>
</table>
Code examples of basic operations

The following instructions give examples of configuring basic data manipulation to remove customer identification numbers associated with the documents.

Console

To configure basic data manipulation to remove custom identification numbers

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. From the left navigation pane, select Document enrichments and then select Add document enrichment.
3. On the Configure basic operations page, choose from the data source that you want to alter document fields and content in.
4. Select the document field name Customer_ID from the dropdown menu, and then select the target action Delete.
5. Select Add basic operation.

AWS CLI

To configure basic data manipulation to remove custom identification numbers

```bash
aws qbusiness create-data-source \
   --name data-source-name \
   --application-id application-id \
   --index-id index-id \
   --role-arn arn:aws:iam::account-id:role/role-name \
   --type S3 \
   --configuration '{"S3Configuration":{"BucketName":"S3-bucket-name"}}' \
   --document-enrichment-configuration '{"InlineDocumentEnrichmentConfiguration": [{"Target":{"key":"Customer_ID", "attributeValueOperator": "DELETE"}}]}'
```
To configure basic data manipulation to remove custom identification numbers

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

qbusiness = boto3.client("qbusiness")

print("Create a data source with customizations")

# Provide the name of the data source
name = "data-source-name"
# Provide the application ID for the data source
application_id = "application-id"
# 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 Document Enrichment
document_enrichment_configuration = {"InlineDocumentEnrichmentConfiguration":
    {
        "Target":{"key":"Customer_ID",
                   "attributeValueOperator": "DELETE"}
    }
}

try:
    data_source_response = qbusiness.create_data_source(
        Name = name,
        ApplicationId = application_id,
        IndexId = index_id,
        RoleArn = role_arn,
        Type = data_source_type
    )
```
Configuration = configuration
DocumentEnrichmentConfiguration = document_enrichment_configuration
)

pprint.pprint(data_source_response)
data_source_id = data_source_response["Id"]

print("Wait for Amazon Q to create the data source with your customizations.")

while True:
    # Get the details of the data source, such as the status
data_source_description = qbusiness.get_data_source(
    DataSourceId = data_source_id,
    ApplicationId = application_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 = qbusiness.start_data_source_sync_job(
    DataSourceId = data_source_id,
    ApplicationId = application_id,
    IndexId = index_id
)

pprint.pprint(sync_response)

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

while True:

    jobs = qbusiness.list_data_source_sync_jobs(
        DataSourceId = data_source_id,
        ApplicationId = application_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

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

print("Program ends.")

Java

To configure basic data manipulation to remove custom identification numbers

```java
package com.amazonaws.qbusiness;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.services.qbusiness.QBusinessClient;
import software.amazon.awssdk.services.qbusiness.model.AttributeValueOperator;
import software.amazon.awssdk.services.qbusiness.model.CreateIndexRequest;
import software.amazon.awssdk.services.qbusiness.model.CreateIndexResponse;
import software.amazon.awssdk.services.qbusiness.model.CreateDataSourceRequest;
import software.amazon.awssdk.services.qbusiness.model.CreateDataSourceResponse;
import software.amazon.awssdk.services.qbusiness.model.DataSourceConfiguration;
import software.amazon.awssdk.services.qbusiness.model.DataSourceStatus;
import software.amazon.awssdk.services.qbusiness.model.DataSourceSyncJob;
import software.amazon.awssdk.services.qbusiness.model.DataSourceSyncJobStatus;
import software.amazon.awssdk.services.qbusiness.model.DataSourceType;
import software.amazon.awssdk.services.qbusiness.model.GetDataSourceRequest;
import software.amazon.awssdk.services.qbusiness.model.GetDataSourceResponse;
import software.amazon.awssdk.services.qbusiness.model.IndexStatus;
import software.amazon.awssdk.services.qbusiness.model.ListDataSourceSyncJobsRequest;
import software.amazon.awssdk.services.qbusiness.model.ListDataSourceSyncJobsResponse;
import software.amazon.awssdk.services.qbusiness.model.DataSourceConfiguration;
import software.amazon.awssdk.services.qbusiness.model.StartDataSourceSyncJobRequest;
import software.amazon.awssdk.services.qbusiness.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 applicationId = "application-id";
    String indexId = "index-id";
    String dataSourceRoleArn = "arn:aws:iam::account-id:role/role-name";
    String s3BucketName = "S3-bucket-name"

    QBusinessClient qbusiness = QBusinessClient.builder().build();

    CreateDataSourceRequest createDataSourceRequest = CreateDataSourceRequest
        .builder()
        .name(dataSourceName)
        .applicationId(applicationId)
        .indexId(indexId)
        .description(experienceDescription)
        .roleArn(experienceRoleArn)
        .type(DataSourceType.S3)
        .configuration(
            DataSourceConfiguration
                .builder()
                .s3Configuration(
                    S3DataSourceConfiguration
                        .builder()
                        .bucketName(s3BucketName)
                        .build()
                ).build()
        ).build()

    .documentEnrichmentConfiguration(
        DocumentEnrichmentConfiguration
            .builder()
            .inlineDocumentEnrichmentConfiguration(Arrays.asList(
                InlineDocumentEnrichmentConfiguration
                    .builder()
                    .target(
                        DocumentAttributeTarget
                            .builder()
                            .key("Customer_ID")
                    )
                    .attributeValueOperator(AttributeValueOperator.DELETE)
                    .build()
            )
    )
}
CreateDataSourceResponse createDataSourceResponse = qbusiness.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 Amazon Q to create the data source %s", dataSourceId));
GetDataSourceRequest getDataSourceRequest = GetDataSourceRequest
  .builder()
  .applicationId(applicationId)
  .indexId(indexId)
  .datasourceId(dataSourceId)
  .build();

while (true) {
  GetDataSourceResponse getDataSourceResponse = qbusiness.getDataSource(getDataSourceRequest);

  DataSourceStatus status = getDataSourceResponse.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()
  .applicationId(applicationId)
  .indexId(indexId)
  .datasourceId(dataSourceId)
  .build();
StartDataSourceSyncJobResponse startDataSourceSyncJobResponse = qbusiness.startDataSourceSyncJob(startDataSourceSyncJobRequest);
System.out.println(String.format("Waiting for the data source to sync with the application %s index %s for execution ID %s", applicationId, indexId, startDataSourceSyncJobResponse.executionId()));
// For this example, there should be one job
ListDataSourceSyncJobsRequest listDataSourceSyncJobsRequest =
ListDataSourceSyncJobsRequest
.builder()
.applicationId(applicationId)
.indexId(indexId)
.datasourceId(dataSourceId)
.build();

while (true) {
    ListDataSourceSyncJobsResponse listDataSourceSyncJobsResponse =
    qbusiness.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");

Using Lambda functions

You can use Lambda functions to prepare your document attributes for advanced data
manipulation. For example, you could use Optical Character Recognition (OCR), which interprets
text from images and treats each image as a textual document. Or, you could retrieve the current
date-time in a specific time zone and then insert the date-time where there's an empty value for a
date field.

You can choose to apply a basic operation first and then use a Lambda function to manipulate your
data, and the reverse.
**Note**

Amazon Q can't create a target document attribute field if it isn't already created as an index field.

---

**Topics**

- Lambda functions using the Amazon Q API
- Lambda functions using the Amazon Q console
- IAM roles for Lambda functions
- Examples of Lambda functions
- Code examples of Lambda functions
- Data contracts for Lambda functions

**Lambda functions using the Amazon Q API**

To apply a Lambda function, you specify your advanced data manipulation logic using the `DocumentEnrichmentConfiguration` object when you use either the `BatchPutDocument` API operation or the `CreateDataSource` operation.

Your Lambda functions must follow the mandatory request and response structures. For more information, see Data contracts for Lambda functions.

Use the following parameters to create your configuration:

- `InlineDocumentEnrichmentConfiguration` – Configuration information to alter document attributes during ingestion.
- `PostExtractionHookConfiguration` – Configuration information to invoke a Lambda function on structured documents with their metadata and text already extracted.
- `PreExtractionHookConfiguration` – Configuration information to invoke a Lambda function on raw documents before metadata and text has been extracted from them.
- `PreExtractionHookConfiguration RoleArn` – The Amazon Resource Name (ARN) of a role under `PreExtractionHookConfiguration` with permissions to run `PreExtractionHookConfiguration` and to access the Amazon S3 bucket when you use `PreExtractionHookConfiguration`. 
• PostExtractionHookConfiguration RoleArn – The Amazon Resource Name (ARN) of a role under PostExtractionHookConfiguration with permissions to run PreExtractionHookConfiguration and to access the Amazon S3 bucket when you use PostExtractionHookConfiguration.

You can configure only one Lambda function for PreExtractionHookConfiguration 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. Your Lambda function for PostExtractionHookConfiguration must not exceed a run time of 1 minute.

You can configure Amazon Q 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 Q invokes a function that inserts the current date-time.

For more information, see the following topics in the Amazon Q API Reference:

• BatchPutDocument
• CreateDataSource
• DocumentEnrichmentConfiguration
• DocumentAttributeCondition

Lambda functions using the Amazon Q console

To configure a Lambda function using the console

1. Select your index, and then select Document enrichments from the navigation menu.
2. To configure Lambda functions, go to Configure Lambda functions.

IAM roles for Lambda functions

When you use the Lambda functions for CDE, you need an IAM role for the following:
• A role for PreExtractionHookConfiguration with permissions to run PreExtractionHookConfiguration and to access the Amazon S3 bucket when you use PreExtractionHookConfiguration.

• A role for PostExtractionHookConfiguration with permissions to run PreExtractionHookConfiguration and to access the Amazon S3 bucket when you use PostExtractionHookConfiguration.

Both AWS Identity and Access Management (IAM) roles must have the permissions to:

• Run PreExtractionHookConfiguration and/or PostExtractionHookConfiguration. To apply advanced alterations of your document metadata and content during the ingestion process, configure a Lambda function for PreExtractionHookConfiguration and/or PostExtractionHookConfiguration.

• (Optional) If you choose to activate Server Side Encryption for your Amazon S3 bucket, you must provide permissions to use the AWS KMS key customer to encrypt and decrypt the objects stored in your Amazon S3 bucket.

A role policy to allow Amazon Q to run PreExtractionHookConfiguration with encryption for your Amazon S3 bucket.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name"
            ]
        }
    ]
}
```
An role policy to allow Amazon Q to run PreExtractionHookConfiguration without encryption.

```json
{
    "Version": "2012-10-17",
    "Statement": [{
        "Action": [
            "s3:GetObject",
            "s3:PutObject"
        ],
        "Resource": ["arn:aws:s3:::bucket-name", "arn:aws:s3:::bucket-name/*"],
        "Effect": "Allow"
    },
    
    "Action": ["s3:ListBucket"
    ]
}
```
A role policy to allow Amazon Q to run 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",
            "arn:aws:s3:::bucket-name/*"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::bucket-name"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "lambda:InvokeFunction"
        ],
        "Resource": "arn:aws:lambda:your-region:your-account-id:function:pre-extraction-lambda-function"
    }
}
```
An role policy to allow Amazon Q to run PostExtractionHookConfiguration without encryption.
We recommend that you include `aws:sourceAccount` and `aws:sourceArn` in the trust policy. Their inclusion limits permissions and securely checks if `aws:sourceAccount` and `aws:sourceArn` are the same values as provided in the IAM role policy for the `sts:AssumeRole` action. This approach prevents unauthorized entities from accessing your IAM roles and their permissions. For more information, see [confused deputy problem](#) in the *IAM User Guide*.
Examples of Lambda functions

This section outlines two examples of using Lambda functions.

Example 1: Extracting text from images to create textual documents

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.

The following table shows data before advanced manipulation is applied.

<table>
<thead>
<tr>
<th>Document_ID</th>
<th>Document_Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>image_1.png</td>
</tr>
<tr>
<td>2</td>
<td>image_2.png</td>
</tr>
<tr>
<td>3</td>
<td>image_3.png</td>
</tr>
</tbody>
</table>

The following table shows data after advanced manipulation is applied.

<table>
<thead>
<tr>
<th>Document_ID</th>
<th>Document_Image</th>
<th>Document_Image_Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>image_1.png</td>
<td>Mailed survey response</td>
</tr>
<tr>
<td>2</td>
<td>image_2.png</td>
<td>Mailed survey response</td>
</tr>
<tr>
<td>3</td>
<td>image_3.png</td>
<td>Mailed survey response</td>
</tr>
</tbody>
</table>

Example 2: Replacing empty values in the Last_Updated field with the current date-time

The following is an example of using a Lambda function to insert the current date-time for empty date values. This example uses the condition that, if a date field value is null, then the value is replaced with the current date-time.

The following table shows data before advanced manipulation is applied.
The following table shows data after advanced manipulation is applied.

<table>
<thead>
<tr>
<th>Document_ID</th>
<th>Body_Text</th>
<th>Last_Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Example text</td>
<td>January 1, 2020</td>
</tr>
<tr>
<td>2</td>
<td>Example text</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Example text</td>
<td>July 1, 2020</td>
</tr>
</tbody>
</table>

**Code examples of Lambda functions**

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. Sign in to the AWS Management Console and open the Amazon Q console at [https://console.aws.amazon.com/amazonq/](https://console.aws.amazon.com/amazonq/).
2. From the left navigation menu, choose **Enhancements**, and then choose **Document enrichments**.
3. In **Document enrichments**, choose **Add document enrichment**.
4. In **Configure basic operations**, for **Document enrichment source**, choose a data source connected to your application.
5. (Optional) To apply basic manipulations to your document fields and content, go to **Configure basic operations** and choose **Next** to save your configuration.
6. On the **Configure Lambda functions** page, in the **Lambda for pre-extraction** section, select your Lambda function ARN and your Amazon S3 bucket using the dropdown menus.

7. To add your IAM access role, select the option to create a new role from the dropdown. This step creates the required Amazon Q permissions to create the document enrichment.

8. Select **Add basic operation**.

**AWS CLI**

**To configure a Lambda function for advanced data manipulation on the raw, original data**

```bash
aws qbusiness create-data-source \
  --name data-source-name \
  --application-id application-id \
  --index-id index-id \
  --role-arn arn:aws:iam::account-id:role/role-name \
  --type S3 \
  --configuration '{"S3Configuration":{"BucketName":"S3-bucket-name"}}' \
  --document-enrichment-configuration '{"InlineDocumentEnrichmentConfiguration": ["Target":{"key":"Customer_ID","attributeValueOperator":true}]}'
```

**Python**

**To configure a Lambda function for advanced data manipulation on the raw, original data**

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

def qbusiness(name, application_id, index_id, role_arn):
    qbusiness = boto3.client("qbusiness")
    print("Create a data source with customizations")

    name = "data-source-name"
    application_id = "application-id"
    index_id = "index-id"
    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 Document Enrichment
configuration = {"S3Configuration":
    {
      "BucketName": S3_bucket_name
    }
}

document_enrichment_configuration = {"InlineDocumentEnrichmentConfiguration":[
    {
      "Target":{"key":"Customer_ID",
                "attributeValueOperator": "DELETE"}
    }
]
}

try:
    data_source_response = qbusiness.create_data_source(
        Name = name,
        ApplicationId = application_id,
        IndexId = index_id,
        RoleArn = role_arn,
        Type = data_source_type
        Configuration = configuration
        DocumentEnrichmentConfiguration = document_enrichment_configuration
    )

    pprint.pprint(data_source_response)

    data_source_id = data_source_response["Id"]

    print("Wait for Amazon Q to create the data source with your customizations.")

    while True:
        # Get the details of the data source, such as the status
        data_source_description = qbusiness.get_data_source(
            DataSourceId = data_source_id,
            ApplicationId = application_id,
            IndexId = index_id
        )
        status = data_source_description["Status"]
        print("Creating data source. Status: "+status)
        time.sleep(60)
        if status != "CREATING":
            break
break

print("Synchronize the data source.")

sync_response = qbusiness.start_data_source_sync_job(
    DataSourceId = data_source_id,
    ApplicationId = application_id,
    IndexId = index_id
)

pprint.pprint(sync_response)

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

while True:

    jobs = qbusiness.list_data_source_sync_jobs(
        DataSourceId = data_source_id,
        ApplicationId = application_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**

```java
package com.amazonaws.qbusiness;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.services.qbusiness.QBusinessClient;
import software.amazon.awssdk.services.qbusiness.model.AttributeValueOperator;
```
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 applicationId = "application-id";
        String indexId = "index-id";
        String dataSourceRoleArn = "arn:aws:iam::account-id:role/role-name";
        String s3BucketName = "S3-bucket-name"

        QBusinessClient qbusiness = QBusinessClient.builder().build();

        CreateDataSourceRequest createDataSourceRequest = CreateDataSourceRequest
                .builder()
                .name(dataSourceName)
                .applicationId(applicationId)
                .indexId(indexId)
                .description(experienceDescription)
                .roleArn(experienceRoleArn)
                .type(DataSourceType.S3)
                .configuration(
                "Using Lambda functions"
DataSourceConfiguration
    .builder()
    .s3Configuration(
        S3DataSourceConfiguration
            .builder()
            .bucketName(s3BucketName)
            .build()
    ).build()}
)
    .documentEnrichmentConfiguration(
        DocumentEnrichmentConfiguration
            .builder()
            .inlineConfigurations(Arrays.asList(
                InlineDocumentEnrichmentConfiguration
                    .builder()
                    .target(
                        DocumentAttributeTarget
                            .builder()
                            .key("Customer_ID")
                    )
                    .attributeValueOperator(AttributeValueOperator.DELETE)
                    .build()
            )).build();

CreateDataSourceResponse createDataSourceResponse = qbusiness.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 Amazon Q to create the data source %s", dataSourceId));
    GetDataSourceRequest getDataSourceRequest = GetDataSourceRequest
        .builder()
        .applicationId(applicationId)
        .indexId(indexId)
        .datasourceId(dataSourceId)
        .build();

    while (true) {
        GetDataSourceResponse getDataSourceResponse = qbusiness.getDataSource(getDataSourceRequest);
DataSourceStatus status = getDataSourceResponse.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()
        .applicationId(applicationId)
        .indexId(indexId)
        .datasourceId(dataSourceId)
        .build();
StartDataSourceSyncJobResponse startDataSourceSyncJobResponse =
    qbusiness.startDataSourceSyncJob(startDataSourceSyncJobRequest);
System.out.println(String.format("Waiting for the data source to sync with the application %s index %s for execution ID %s", applicationId, indexId, startDataSourceSyncJobResponse.executionId()));

    // For this example, there should be one job
    ListDataSourceSyncJobsRequest listDataSourceSyncJobsRequest =
        ListDataSourceSyncJobsRequest.builder()
            .applicationId(applicationId)
            .indexId(indexId)
            .datasourceId(dataSourceId)
            .build();

    while (true) {
        ListDataSourceSyncJobsResponse listDataSourceSyncJobsResponse =
            qbusiness.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;
        }
    }
Data contracts for Lambda functions

Lambda functions for advanced data manipulation interact with Amazon Q 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 Q produces an error. Your Lambda function for PreExtractionHookConfiguration should use the following request structure:

```json
{
   "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 DocumentAttribute structure, is as follows:

```json
{
   "attributes": [<DocumentAttribute>]
}
```

DocumentAttribute

```json
{
   "name": <str>,
   "value": <DocumentAttributeValue>
}
```

DocumentAttributeValue

```json
{
   "stringValue": <str>,
   "integerValue": <int>,
   "longValue": <long>,
   "stringListValue": list<str>,
   "dateValue": <str>
}
```
Your Lambda function for PreExtractionHookConfiguration must adhere to the following response structure:

```json
{
    "version": <str>,
    "dataBlobStringEncodedInBase64": <str>, //In the case of a data blob
    "s3ObjectKey": <str>, //In the case of an S3 bucket
    "metadataUpdates": [<DocumentAttribute>]
}
```

Your Lambda function for PostExtractionHookConfiguration should expect the following request structure:

```json
{
    "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": [<DocumentAttribute>]
}
```

Amazon Q uploads your structured document to the specified Amazon S3 bucket. The structured document follows this format:

```
QBusiness document
{
    "textContent": <TextContent>
}
```
Examples of Lambda functions that adhere to data contracts

This section provides examples of how to structure your Lambda functions that adhere to Amazon Q data contracts.

Example 1: A Lambda function that applies advanced manipulation to raw documents

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.

The following code example shows the case of the body content residing in an Amazon S3 bucket

```python
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_DE = s3.get_object(Bucket = s3_bucket, Key = s3_object_key)
    content_before_DE = content_object_before_DE["Body"].read().decode("utf-8")
    content_after_DE = "DEInvolved " + content_before_DE

    # 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_qbusiness_document", Body=json.dumps(content_after_DE))
    return {
```
Example 2: A Lambda function that applies advanced manipulation to structured or parsed documents

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.

```python
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")

    qbusiness_document_object = s3.get_object(Bucket = s3_bucket, Key = s3_key)
    qbusiness_document_string = str
    qbusiness_document_object['Body'].read().decode('utf-8')
    qbusiness_document = json.loads(qbusiness_document_string)
    qbusiness_document["textContent"]['documentBodyText'] = "Changing document body to a short sentence."

    s3.put_object(Bucket = s3_bucket, Key = "dummy_updated_qbusiness_document", Body=json.dumps(qbusiness_document))
```
Example 3: Body content residing in a data blob

```python
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"]}}
        ]
    }
```
Configuring plugins with Amazon Q

You can create and configure plugins for your Amazon Q application. Once configured, plugins can support actions—or write operations and instructions—that can help you boost end user productivity. End users can perform specific tasks related to third-party services from within their web experience chat such as creating a Jira ticket.

For example, your end user might be an IT representative whose Amazon Q chat requires the follow-up action of opening an incident in ServiceNow. They can request that Amazon Q create an incident in ServiceNow on their behalf without leaving their chat.

Currently, Amazon Q supports the following plugins and actions:

- **Jira** – Creating an issue
- **Salesforce** – Creating a case
- **ServiceNow** – Creating an incident
- **Zendesk** – Creating a ticket

Each Amazon Q plugin currently supports a single action. Each Amazon Q application can have up to 4 enabled plugins. No two plugins can be of the same type. Once activated, you can choose to deactivate, reactivate, edit, and delete plugins at any time. You can’t currently customize plugins.

⚠️ **Important**

Once configured, all authorized Amazon Q web experience end users can use plugins to perform supported actions. Currently, end user access to plugins can’t be customized.

**Topics**

- [Configuring a Jira plugin](#)
- [Configuring a Salesforce plugin](#)
- [Configuring a ServiceNow plugin](#)
- [Configuring a Zendesk plugin](#)
- [Using Amazon Q plugins](#)
- [Managing Amazon Q plugins](#)
Configuring a Jira plugin

Jira is a project management tool that creates issues (tickets) for software development, product management, and bug tracking. If you're a Jira user, you can create an Amazon Q plugin to allow your end users to create Jira issues from within their web experience chat.

To create a Jira plugin, you need configuration information from your Jira instance to set up a connection between Amazon Q and Jira and allow Amazon Q to perform actions in Jira.

For more information on how to use plugins during your web experience chat, see Using plugins.

Topics

- Prerequisites
- Service access roles
- Creating a plugin

Prerequisites

Before you configure your Amazon Q Jira plugin, you must do the following:

- Set up a new user in your Jira instance with scoped permissions for performing actions in Amazon Q.
- (Optional) Create an API token for the new user that you created.
- Note this user's Jira username and Jira account password (and optionally, their API token). You will need this basic authentication information for creating an AWS Secrets Manager secret during the plugin configuration process.
- Note the base URL of your Jira Cloud instance hosted by Atlassian. For example: https://yourcompany.atlassian.net.

Service access roles

To successfully connect Amazon Q to Jira, you need to give Amazon Q the following permission to access your Secrets Manager secret to get your Jira credentials. Amazon Q assumes this role to access your Jira credentials.

The following is the service access IAM role required:
If you use the console and choose to create a new IAM role, Amazon Q creates the role for you. If you use the console and choose to use an existing secret, or you use the API, make sure your IAM role contains these permissions.

Creating a plugin

To create a Jira plugin for your web experience chat, you can use the AWS Management Console or the `CreatePlugin` API operation. The following tabs provide a procedure to create a Jira plugin using the console and code examples for the AWS CLI.

Console

To create a Jira plugin

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In Applications, select the name of your application from the list of applications.
3. From the left navigation menu, choose Enhancements, and then choose Plugins.
4. For Plugins, choose Add plugin.
5. For Add plugins, choose Jira.
6. For Jira, enter the following information:

   a. **Name, Plugin name** – A name for your Amazon Q plugin. The name can include hyphens (-), but not spaces, and can have a maximum of 1,000 alphanumeric characters.
b. **Service access** – Choose **Create and add a new service role** or **Use an existing service role**. Make sure that your service role has the necessary permissions.

c. **URL** – The base URL of your Jira Cloud instance hosted by Atlassian. For example: https://yourcompany.atlassian.net.

d. **Authentication** – Choose to **Create and add a new secret** or **Use an existing one**.

   If you choose to create a new secret, a Secrets Manager secret window opens requesting the following information:

   i. **Secret name** – A name for your Secrets Manager secret.

   ii. **Jira username** – The username for your Jira user.

   iii. **Jira password/API token** – The password/API token for your Jira user.

7. **Tags – optional** – Add an optional tag to track your plugin.

8. Choose **Save**.

**AWS CLI**

**To create a Jira plugin**

```bash
aws qbusiness create-plugin \ 
--application-id application-id \ 
--display-name display-name \ 
--type JIRA \ 
--server-url https://example.atlassian.net \ 
--auth-configuration basicAuthConfiguration="{secretArn=<secret-arn>,roleArn=<role-arn>}"`
```

**Configuring a Salesforce plugin**

Salesforce is a customer relationship management (CRM) tool for managing support, sales, and marketing teams that you can use to create cases (tickets) to track issues. If you’re a Salesforce user, you can create an Amazon Q plugin to allow your end users to create Salesforce cases from within their web experience chat.
To create a Salesforce plugin, you need configuration information from your Salesforce instance to set up a connection between Amazon Q and Salesforce and allow Amazon Q to perform actions in Salesforce.

For more information on how to use plugins during your web experience chat, see Using plugins.

Topics

- Prerequisites
- Service access roles
- Creating a plugin

Prerequisites

Before you configure your Amazon Q Salesforce plugin, you must do the following:

- Set up a Connected App using the admin role in your Salesforce instance with Client Credentials Flow enabled.
- As an admin, configure an execution user with scoped permissions for performing actions in Amazon Q. For instructions, see Configure a Connected App for the OAuth 2.0 Client Credentials Flow in the Salesforce documentation.
- Note your Salesforce Connected App's consumer key (client_id) and your Salesforce Connected App Consumer secret (client_secret). You will need this Oauth 2.0 authentication information for creating an AWS Secrets Manager secret during the plugin configuration process.
- Note the Salesforce My Domain URL of your Salesforce organization. For example: https://yourdomain.my.salesforce.com.

Service access roles

To successfully connect Amazon Q to Salesforce, you need to give Amazon Q the following permission to access your Secrets Manager secret to get your Salesforce credentials. Amazon Q assumes this role to access your Salesforce credentials.

The following is the service access IAM role required:

```json
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
```
If you use the console and choose to create a new IAM role, Amazon Q creates the role for you. If you use the console and choose to use an existing secret, or you use the API, make sure your IAM role contains these permissions.

Creating a plugin

To create a Salesforce plugin for your web experience chat, you can use the AWS Management Console or the `CreatePlugin` API operation. The following tabs provide a procedure for creating a Salesforce plugin using the console and code examples for the AWS CLI.

Console

**To create a Salesforce plugin**

1. Sign in to the AWS Management Console and open the Amazon Q console at [https://console.aws.amazon.com/amazonq/](https://console.aws.amazon.com/amazonq/).
2. From the Amazon Q console, in **Applications**, select the name of your application from the list of applications.
3. From the left navigation menu, choose **Enhancements**, and then choose **Plugins**.
4. For **Plugins**, choose **Add plugin**.
5. For **Add plugins**, choose **Salesforce**.
6. For **Salesforce**, enter the following information:
   a. **Name**, for **Plugin name** – A name for your Amazon Q plugin. The name can include hyphens (-), but not spaces, and can have a maximum of 1,000 alphanumeric characters.
   b. **Service access** – Choose **Create and add a new service role** or **Use an existing service role**. Make sure that your service role has the necessary permissions.
c. **URL** – My Domain URL of your Salesforce organization. For example: https://yourdomain.my.salesforce.com

d. **Authentication** – Choose **Create and add a new secret** or **Use an existing one**. Your secret must contain the following information:

i. **Secret name** – A name for your Secrets Manager secret.

ii. **Connected app consumer key** – The consumer key for your Salesforce connected app.

iii. **Connected app consumer secret** – The consumer secret for your Salesforce connected app.

7. **Tags** – **optional** – An optional tag to track your plugin.

8. Choose **Save**.

**AWS CLI**

**To create a Salesforce plugin**

```bash
aws qbusiness create-plugin \
--application-id application-id \ 
--display-name display-name \ 
--type SALESFORCE \ 
--server-url //example.my.salesforce.com \ 
--auth-configuration OAuth2ClientCredentialConfiguration="{secretArn=<secret-arn>,roleArn=<role-arn}>"
```

**Configuring a ServiceNow plugin**

ServiceNow provides a cloud-based service management system to create and manage organization-level workflows, such as IT services, ticketing systems, and support. ServiceNow uses incidents (tickets) to track issues. If you’re a ServiceNow user, you can create an Amazon Q plugin to allow your end users to create ServiceNow cases from within their web experience chat.

To create a ServiceNow plugin, you need configuration information from your ServiceNow instance to set up a connection between Amazon Q and ServiceNow and allow Amazon Q to perform actions in ServiceNow.
For more information on how to use plugins during your web experience chat, see Using plugins.

Topics
- Prerequisites
- Service access roles
- Creating a plugin

Prerequisites

Before you configure your Amazon Q ServiceNow plugin, you must do the following:

- As an admin, set up a new user in your ServiceNow instance with scoped permissions for performing actions in Amazon Q.
- Note your ServiceNow username and ServiceNow password. You will need this basic authentication information for creating an AWS Secrets Manager secret during the plugin configuration process.
- Note the base URL of your ServiceNow instance. For example: https://yourinstance.service-now.com.

Service access roles

To successfully connect Amazon Q to ServiceNow, you need to give Amazon Q the following permission to access your Secrets Manager secret to get your ServiceNow credentials. Amazon Q assumes this role to access your ServiceNow credentials.

The following is the service access IAM role required:

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": ["secretsmanager:GetSecretValue"],
         "Resource": [
            "arn:aws:secretsmanager:{{your-region}}:{{your-account-id}}:secret:[[secret-id]]"
         ]
      }
   ]
}
```
If you use the console and choose to create a new IAM role, Amazon Q creates the role for you. If you use the console and choose to use an existing secret, or you use the API, make sure your IAM role contains these permissions.

Creating a plugin

To create a ServiceNow plugin for your web experience chat, you can use the AWS Management Console or the CreatePlugin API operation. The following tabs provide a procedure for creating a ServiceNow plugin using the console and code examples for the AWS CLI.

Console

To create a ServiceNow plugin

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. From the Amazon Q console, in Applications, select the name of your application from the list of applications.
3. From the left navigation menu, choose Enhancements, and then choose Plugins.
4. For Plugins, choose Add plugin.
5. For Add plugins, choose ServiceNow.
6. For ServiceNow, enter the following information:
   a. Name, for Plugin name – A name for your Amazon Q plugin. The name can include hyphens (-), but not spaces, and can have a maximum of 1,000 alphanumeric characters.
   b. Service access – Choose Create and add a new service role or Use an existing service role. Make sure that your service role has the necessary permissions.
   c. URL – The base URL of your ServiceNow instance. For example: https://yourinstance.service-now.com
   d. Authentication – Choose Create and add a new secret or Use an existing one. Your secret must contain the following information:
      i. Secret name – A name for your Secrets Manager secret.
ii. **ServiceNow username** – The username for your ServiceNow user.

iii. **ServiceNow password** – The password for your ServiceNow user.

7. **Tags – optional** – An optional tag to track your plugin.

8. Choose **Save**.

**AWS CLI**

**To create a ServiceNow plugin**

```
aws qbusiness create-plugin
--application-id application-id
--display-name display-name
--type SERVICE-NOW
--server-url //example.service-now.com
--auth-configuration basicAuthConfiguration="{secretArn=<secret-arn>,roleArn=<role-arn>}
```

**Configuring a Zendesk plugin**

Zendesk is a customer relationship management system that helps businesses automate and enhance customer support interactions by creating tickets to track work. If you're a Zendesk user, you can create an Amazon Q plugin to allow your end users to create Zendesk cases from within their web experience chat.

To create a Zendesk plugin, you need configuration information from your Zendesk instance to set up a connection between Amazon Q and Zendesk and allow Amazon Q to perform actions in Zendesk.

For more information on how to use plugins during your web experience chat, see [Using plugins](#).

**Topics**

- [Prerequisites](#)
- [Service access roles](#)
- [Creating a plugin](#)
Prerequisites

Before you configure your Amazon Q Zendesk plugin, you must do the following:

- As an admin, set up a new user in your Zendesk instance with scoped permissions for performing actions in Amazon Q.
- (Optional) Create an API token for that new user.
- Note your Zendesk username and Zendesk password/API token. You will need this basic authentication information for creating an AWS Secrets Manager secret during the plugin configuration process.
- Note the base URL of your Zendesk instance. For example: https://yoursubdomain.zendesk.com.

Service access roles

To successfully connect Amazon Q to Zendesk, you need to give Amazon Q the following permission to access your Secrets Manager secret to get your Zendesk credentials. Amazon Q assumes this role to access your Zendesk credentials.

The following is the service access IAM role required:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:{{your-region}}:{{your-account-id}}:secret:[[secret-id]]"
            ]
        }
    ]
}
```

If you use the console and choose to create a new IAM role, Amazon Q creates the role for you. If you use the console and choose to use an existing secret, or you use the API, make sure your IAM role contains these permissions.
Creating a plugin

To create a Zendesk plugin for your web experience chat, you can use AWS Management Console or the `CreatePlugin` API operation. The following tabs provide a procedure for creating a Zendesk plugin using the console and code examples for the AWS CLI.

Console

**To create a Zendesk plugin**

1. Sign in to the AWS Management Console and open the Amazon Q console at [https://console.aws.amazon.com/amazonq/](https://console.aws.amazon.com/amazonq/).
2. From the Amazon Q console, in **Applications**, select the name of your application from the list of applications.
3. From the left navigation menu, choose **Enhancements**, and then choose **Plugins**.
4. For **Plugins**, choose **Add plugin**.
5. For **Add plugins**, choose **Zendesk**.
6. For **Zendesk**, enter the following information:
   a. **Name, Plugin name** – A name for your Amazon Q plugin. The name can include hyphens (-), but not spaces, and can have a maximum of 1,000 alphanumeric characters.
   b. For **Service access** – Choose **Create and add a new service role** or **Use an existing service role**. Make sure that your service role has the necessary permissions.
   c. **URL** – The base URL of your Zendesk instance. For example: [https://yoursubdomain.zendesk.com](https://yoursubdomain.zendesk.com)
   d. **Authentication** – Choose **Create and add a new secret** or **Use an existing one**. Your secret must contain the following information:
      i. **Secret name** – A name for your Secrets Manager secret.
      ii. **Zendesk username** – The username for your Zendesk user.
      iii. **Zendesk password/API token** – The password/API token for your Zendesk user.
7. **Tags** – **optional** – An optional tag to track your plugin.
8. Choose **Save**.
AWS CLI

**To create a Zendesk plugin**

```bash
aws qbusiness create-plugin \
--application-id application-id \
--display-name display-name \
--type ZENDESK \
--server-url //example.zendesk.com \
--auth-configuration basicAuthConfiguration="{secretArn=<secret-arn>,roleArn=<role-arn>"
```

**Using Amazon Q plugins**

After plugins have been configured, you can use them to perform supported actions in your Amazon Q web experience chat. This topic provides an overview of how to use plugins.

⚠️ **Important**

Once configured, all authorized Amazon Q web experience end users can use plugins to perform supported actions. Currently, end user access to plugins can't be customized.

**Topics**

- [User flow for performing a plugin action](#)
- [Example plugin action prompts](#)

**User flow for performing a plugin action**

The following describes the user flow for performing an action from within a web experience chat:

1. Navigate to the deployed web experience URL and sign with your credentials on the login screen.
2. You can choose to enact plugin actions in two ways:
   - Ask to perform an action directly. For example: Create a Jira ticket for a broken mouse. See [Quick create](#) for more details.
• Start chatting in your web experience to find answers to your questions. Then choose to include the conversation context in any plugin action that you take. For example: Summarize this conversation and create a Jira ticket. For more information, see Contextual create.

3. In response to your prompt for an action, Amazon Q displays a review form where you fill in the necessary information required to successfully complete an action.

4. To successfully complete the action, you need to submit it. Your web experience will display a success message if the action succeeds, or an error message if the action fails.

⚠️ Note

If the request to create an action is successful, you can view ticket created as a result of the action.

Example plugin action prompts

There are two ways you can choose to use plugins in your web experience chat, quick creation and contextual creation.

Topics

• Quick create
• Contextual create

Quick create

Using quick creation you can directly instruct your web experience to perform a plugin action. For example:

• Create a Zendesk ticket for a broken mouse
• Log an incident in ServiceNow for network outage
• Cut an issue in Jira for a broken link on a web page
• Create a Salesforce case for a missing invoice
Contextual create

Using contextual creation you can include conversation contexts to create tickets. For example, consider the following example conversation flows:

Example contextual create actions

• Example 1: Create a ServiceNow incident
• Example 2: Create a ZenDesk ticket
• Example 3: Create a Salesforce case
• Example 4: Create a Jira issue

Example 1: Create a ServiceNow incident

• User prompt 1 – How to resolve network issues
• Amazon Q response – Sample response
• User prompt 2 – How to reset my router
• Amazon Q response – Sample response
• User action request – Summarize this conversation and create a ServiceNow incident

Example 2: Create a ZenDesk ticket

• User prompt 1 – Compare Amazon Kendra with OpenSearch
• Amazon Q response – Sample response
• User action request – Create a Zendesk ticket to migrate to Amazon Kendra

Example 3: Create a Salesforce case

• User prompt 1 – Where is the IT office located
• Amazon Q response – Sample response
• User prompt 2 – What floor is the office located in
• Amazon Q response – Sample response
• User action request – Create a case in Salesforce summarizing this conversation
Example 4: Create a Jira issue

- **User prompt 1** – How do I enable auto-scaling in EC2
- **Amazon Q response** – *Sample response*
- **User prompt 2** – How do I create an auto-scaling group
- **Amazon Q response** – *Sample response*
- **User action request** – Summarize this conversation and create an issue in Jira

Managing Amazon Q plugins

To manage Amazon Q plugins, you can take the following actions:

**Actions**

- [Updating a plugin](#)
- [Deleting a plugin](#)
- [Getting plugin properties](#)
- [Listing plugins](#)

**Updating a plugin**

To update a plugin, you can use AWS Management Console or the `UpdatePlugin` API operation. The following tabs provide a procedure for the console and code examples for the AWS CLI.

**Console**

**To update a plugin**

1. Sign in to the AWS Management Console and open the Amazon Q console.
2. From the Amazon Q console, in **Applications**, select the name of your application from the list of applications.
3. From the left navigation menu, choose **Enhancements**, and then choose **Plugins**.
4. For **Plugins**, select the plugin that you want to update, and then choose **Actions**.
5. For **Actions**, choose **Edit**.

On the plugins configuration page, you can edit your settings.
To deactivate a plugin

1. Sign in to the AWS Management Console and open the Amazon Q console.
2. From the Amazon Q console, in **Applications**, select the name of your application from the list of applications.
3. From the left navigation menu, choose **Enhancements**, and then choose **Plugins**.
4. For **Plugins**, select the plugin that you want to deactivate, and then choose **Actions**.
5. For **Actions**, choose **Deactivate**.

   Your plugin will be deactivated. After your plugin is deactivated, its status will change to **Inactive**.

To reactivate a plugin

1. Sign in to the AWS Management Console and open the Amazon Q console.
2. From the Amazon Q console, in **Applications**, select the name of your application from the list of applications.
3. From the left navigation menu, choose **Enhancements**, and then choose **Plugins**.
4. For **Plugins**, select the plugin that you want to reactivate, and then choose **Actions**.
5. For **Actions**, choose **Reactivate**.

   Your plugin will be activated. After your plugin is reactivated, its status will change to **Active**.

AWS CLI

To edit a plugin

```bash
aws qbusiness update-plugin
   --application-id application-id
   --plugin-id plugin-id
   --display-name display-name
   --server-url https://example.atlassian.net
   --auth-configuration basicAuthConfiguration="{secretArn=<secret-arn>,roleArn=<role-arn}>"
```
**To disable a plugin**

```
aws qbusiness update-plugin \
  --application-id application-id \
  --plugin-id plugin-id \
  --state DISABLED
```

**To enable a plugin**

```
aws qbusiness update-plugin \
  --application-id application-id \
  --plugin-id plugin-id \
  --state ENABLED
```

**Deleting a plugin**

To delete a plugin, you can use the AWS Management Console or the `DeletePlugin` API operation. The following tabs provide a procedure for the console and code examples for the AWS CLI.

**Console**

**To delete a plugin**

1. Sign in to the AWS Management Console and open the Amazon Q console.
2. From the Amazon Q console, in **Applications**, select the name of your application from the list of applications.
3. From the left navigation menu, choose **Enhancements**, and then choose **Plugins**.
4. For **Plugins**, select the plugin that you want to delete, and then choose **Actions**.
5. For **Actions**, choose **Delete**.
6. In the dialog box, type `delete` to confirm your action.

   The console displays a successful deletion message when the plugin deletion process is finished.
AWS CLI

To delete a plugin

```bash/aws qbusiness delete-plugin \ 
--application-id application-id \ 
--plugin-id plugin-id
```

Getting plugin properties

To get the details of an Amazon Q plugin, you can use either the AWS Management Console or the GetPlugin API operation. The following tabs provide a procedure for the console and code examples for the AWS CLI.

Console

**To get plugin details**

1. Sign in to the AWS Management Console and open the Amazon Q console.
2. From the Amazon Q console, in **Applications**, select the name of your application from the list of applications.
3. From the left navigation menu, choose **Enhancements**, and then choose **Plugins**.
4. For **Plugins**, select the configured plugin that you want to see details for.
5. On the **Plugin settings** page, the following details are available:
   - **Name** – The name of your plugin.
   - **Type** – The type of your plugin.
   - **AWS Secrets Manager** – The Secrets Manager secret.
   - **Creation time** – The time stamp for when your plugin was created.
   - **Plugin ID** – The ID that's assigned to your plugin.

AWS CLI

**To get plugin details**
Listing plugins

To list Amazon Q plugins, you can use the AWS Management Console or the ListPlugins API operation. The following tabs provide a procedure for the console and code examples for the AWS CLI.

Console

To list plugins

1. Sign in to the AWS Management Console and open the Amazon Q console.
2. From the Amazon Q console, in Applications, select the name of your application from the list of applications.
3. From the left navigation menu, choose Enhancements, and then choose Plugins.
4. In Plugins, a list of plugins that are attached to your application is available.

AWS CLI

To list plugins

```
aws qbusiness list-plugins \
--application-id application-id
```

Admin controls and guardrails in Amazon Q

With Amazon Q, you can customize your application to your organizational needs. Amazon Q offers application guardrails or chat controls that you can configure to control the end user chat experience.

Using the guardrails feature, you can define global controls and topic-level controls for your application like the following:

- Control whether end users can upload files in chat to generate responses from uploaded files.
• Specify that all Amazon Q chat responses are generated only using enterprise data.
• Specify that your application can also use the model knowledge of its underlying large language model (LLM) to generate responses when it can’t find answers in your enterprise data.
• Control how Amazon Q responds to specific topics in chat.
• Customize which users and groups Amazon Q topic-level controls apply to.

Topics

• Key terms for Amazon Q guardrails and chat controls
• Using global controls in Amazon Q
• Using topic-level controls in Amazon Q
• Managing Amazon Q admin controls and guardrails

Key terms for Amazon Q guardrails and chat controls

The following are key terms you should know to understand guardrails in Amazon Q:

• **Enterprise data** – Data connected to your application using either an Amazon Q connector, direct document upload, or through an Amazon Kendra retriever.
• **Model knowledge** – The underlying knowledge outside your enterprise data that your large language model (LLM) is trained on.
• **Topic** – An admin user defined natural language topic.
• **Global controls** – Application level controls for controlling the sources that your application uses to generate responses (model knowledge and enterprise data, or enterprise data only). Global controls also define and control blocked phrases within your application.
• **Topic controls** – Topic-specific controls to determine the web application's behavior when it encounters a mention of a blocked topic by an end user.
• **Rules** – An application behavior logic configured to manage a controlled topic for a particular group of users.

Using global controls in Amazon Q

You can use Amazon Q global controls to configure settings that apply to conversations in your application.
The following are the global features that you can customize:

**Global controls**
- **Response settings**
- **Blocked phrases**
- **Feature control**
- **Customizing global controls**

### Response settings

Use response settings to prevent Amazon Q from using large language model (LLM) model knowledge and limit responses to enterprise data. Use this feature to decide whether chat response generation will be restricted to enterprise content, or if Amazon Q will use its model knowledge to generate responses when it can't find answers in your enterprise data.

By default, your application is set to produce responses using only enterprise data. You can choose to update this setting.

**Important**

Displaying sample prompts to your end user using the Amazon Q [Quick prompts](#) feature might not work if you choose to restrict response generation to enterprise data.

Global controls apply to all supported conversation interactions, except when it conflicts with a specific topic-level control. In that case, a topic-level control takes precedence.

### Blocked phrases

You can define blocked phrases for your application. Amazon Q won't respond to any chat questions that include these words or mention these words in any responses. You can choose up to 20 words.
Additionally, you can optionally configure a custom message to be displayed to your end users in response to any mention of blocked phrases during chat. You can use this message to inform them that word is blocked and provide them with further guidance on next steps.

By default, your application doesn't define any blocked words. You can choose to add these words when you edit and update your global control guardrails.

**Feature control**

You can control whether end users can upload files during chat to ask questions based on the uploaded document.

By default, your application allows your end users to directly upload files in chat.

**Customizing global controls**

When you create an Amazon Q application, it's assigned the following default global controls:

- Generate responses from enterprise data only.
- No blocked words allowed.
- File upload by end user during chat is activated.

To update global topic controls for your web experience chat, you can use the AWS Management Console or the `UpdateChatControlsConfiguration` API operation. The following tabs provide a procedure for the console and code examples for the AWS CLI.

> **Note**
> You can't create or delete guardrail global controls. You can only update existing global controls in your application.

**Console**

**To update a global control guardrail**

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. From the left navigation menu, choose Enhancements, and then choose Guardrails.
3. In Guardrails, from Global controls, choose Edit.
4. In Application guardrails, do the following:

- For **Response settings**, **Only produce responses from retrieval augmented generation (RAG)** – Clear the check box if you want to generate responses from your application's LLM world knowledge. The default is to restrict responses to enterprise data.

- For **Blocked words** – Define blocked words for the application. The application will not respond to questions that contain these words or mention them in any responses.

- For **Messaging shown for blocked words** – Choose to create a custom response for your end users informing them of blocked word usage and any next steps to take.

5. For **Feature settings**, choose whether your end users will be allowed to upload files directly in chat to ask questions based on file content.

6. Choose **Save**.

**AWS CLI**

**To update a global control guardrail**

```bash
aws qbusiness update-chat-controls-configuration \
  --application-id application-id \
  --blocked-phrases-configuration-update '{"blockedPhrasesToCreateOrUpdate": ["example phrase 1", "example phrase 2"], "blockedPhrasesToDelete": ["example phrase 1", "example phrase 2"], "systemMessageOverride": "user facing message when blocked phrase encountered"}' \
  --client-token clientToken \
  --response-scope ENTERPRISE_CONTENT_ONLY | EXTENDED_KNOWLEDGE_ENABLED
```

**Using topic-level controls in Amazon Q**

You can use topic-level controls to specify special topics within your application. You can configure rules to customize how Amazon Q should respond when a chat message matches a special topic. To streamline your application's response, you provide a name and a short description for how the large language model (LLM) should respond based on the topic-specific guardrail you're building. You can configure up to 10 topic-level controls.

Topic-level controls provide fine-grained customization for your application. For example, you can define a global control guardrail that allows your application to generate responses using model
knowledge. You can also use a content retrieval rule to limit response generation for specific topics to enterprise content.

The following are the topic-level guardrails that you can customize:

**Topic level guardrails**
- **LLM prompt control**
- **Application behavior rules**
- **Creating topic controls**

**LLM prompt control**

You can add up to 5 representative messages that you expect end users to submit about this topic. You can also configure natural language descriptions to define the boundaries of the topic. Amazon Q uses these messages to check the responses that it generates for restricted content.

**Application behavior rules**

You can configure behavior rules that control how Amazon Q responds for each special topic that you specify.

**Note**

You can specify up to 5 rules per special topic.

**Rules**
- **Answer using enterprise data**
- **Blocking special topics**

**Answer using enterprise data**

When your application encounters a special topic, you can choose to allow it to answer from your enterprise data. If you allow responses from your enterprise data, you can further restrict which data sources in your application that your responses are generated from.

You can also choose to specify the specific users or groups within your application to apply this rule to, using either an inclusion logic or an exclusion logic. You can’t use both kinds of logic at
once. If a user is a member of a group with conflicting rules defined, Amazon Q will apply the more restrictive rule to that user.

**Blocking special topics**

When your application encounters a special topic, you can choose to block responses completely. If you do so, you can configure a custom message to display to your end users in response to any mention of blocked words during chat. Use this message to inform your end users that the topic is blocked and provide them with further guidance on next steps.

You can also choose to specify the specific groups within your application to apply this rule to, using either an inclusion logic or an exclusion logic. You can’t use both kinds of logic at once. If a user is a member of a group with conflicting rules defined, Amazon Q will apply the more restrictive rule to that user.

Not specifying an inclusion or exclusion logic will result in the rule being applied to all users.

**Creating topic controls**

To create an Amazon Q topic-level control for your web experience chat, you can use AWS Management Console or the `UpdateChatControlConfiguration` operation. The following tabs provide a procedure for the console and code examples for the AWS CLI.

**Console**

**To create a topic control**

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. From the left navigation menu, choose **Enhancements**, and then choose **Guardrails**.
3. For **Guardrails**, from **Topic specific controls**, choose **Create topic control**.
4. For **Create topic specific controls**, enter the following information:
   - **Name** – Enter a name for your topic-specific control.
   - **Description** – A natural language description for your topic control configuration. Use this to help the LLM better identify queries associated with the topic control you’re configuring.
5. For **Example chat messages**, enter representative phrases that you expect a user to type to invoke this topic. You can add up to 5 messages.
6. (Optional) To configure a rule, choose **Add new rule**.

7. For **Rule 1**, enter the following information:
   
   - In **Behavior in response to guardrail**, for **Behavior** – Choose how Amazon Q will respond to blocked topics: **Answer using enterprise data** or **Block completely**.
   
   - If you choose **Block completely** – Choose to include a custom message to inform your end user of restricted topics from chat and suggest follow up actions.
   
   - If you choose **Answer using enterprise data**, **Data source requirements** – Choose data sources that Amazon Q will use to generate responses.

8. For **User handling**, specify the users or groups that this topic control rule applies to and any users or groups that are exempt from this rule.

9. Choose **Save**.

---

**AWS CLI**

**To create a topic control**

```sh
aws qbusiness update-chat-controls-configuration \
   --application-id application-id \
   --client-token clientToken \
   --topic-configurations-to-create-or-update \
   '[["name":"name","description":"description","exampleChatMessages": \
   ["message1", "message2"],"rules":{{"includedUsersAndGroups":{"userIds": \
   ["userId1","userId2"],"userGroups": ["userGroup1","userGroup2"]},"ruleType": \
   "CONTENT_BLOCKER_RULE","ruleConfiguration":{"contentBlockerRule": \
   {"systemMessageOverride":"custom_message"}}},"excludedUsersAndGroups": \
   {"userIds": ["id1", "id2"],"userGroups": ["group1","group2"]}, "ruleType": \
   "CONTENT_RETRIEVAL_RULE","ruleConfiguration":{"contentRetrievalRule": \
   {"eligibleDataSources": ["indexId":"index-id1","dataSourceId":"data-source-id1"], \
   ["indexId":"index-id2","dataSourceId":"data-source-id2"]}}}}]' \
   --topic-configurations-to-delete \
   '{"name":"existing-topic-name"}'
```

---

**Note**

The user IDs you add to configure topic controls must already exist in your Identity Provider (IdP). You are responsible for validating any user groups you add.
Managing Amazon Q admin controls and guardrails

To manage Amazon Q admin controls and guardrails, you can take the following actions:

Note

You can't create or delete guardrail global controls. You can only update existing global controls in your application.

Actions

- Deleting topic controls
- Getting topic control properties

Deleting topic controls

To delete configured chat controls, you can use AWS Management Console or the `DeleteChatControlsConfiguration` API operation. The following tabs provide a procedure for the console and code examples for the AWS CLI.

Console

To delete topic controls

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. From the left navigation menu, choose Enhancements, and then choose Guardrails.
3. In Guardrails, from Topic specific controls, choose the topic control you want to delete, and then choose Delete.
4. In the dialog box, type delete to confirm your action.

The console displays a successful deletion message when the plugin deletion process is finished.

AWS CLI

To delete a topic specific control
Getting topic control properties

To get the details of Amazon Q topic controls, you can use either the AWS Management Console or the GetChatControlsConfiguration API operation. The following tabs provide a procedure for the console and code examples for the AWS CLI.

Console

To get configured details for admin controls and guardails

1. Sign in to the AWS Management Console and open the Amazon Q console.
2. From the Amazon Q console, in Applications, select the name of your application from the list of applications.
3. From the left navigation menu, choose Enhancements, and then choose Admin controls and guardails.
   
   You will find the details of your configured Global controls and Topic specific controls on the page.

AWS CLI

To get admin controls and guardails details

```
aws qbusiness get-chat-control-configuration \\n--application-id application-id
```
Previewing and customizing an Amazon Q web experience

After creating and enhancing your Amazon Q application, you can preview the Amazon Q web experience that you created for your end users in the AWS console. By previewing your web experience, you can test the features and enhancements that you configured for it.

### Note

You can run a limited number of chat queries from the web experience preview. Only public documents ingested in your index are accessible—and used for generating responses—in the preview. Documents with access control are not accessible in, or searchable from, the preview.

You can customize and preview a web experience by using either the AWS Management Console or the Amazon Q API. If you use the API, previewing your Amazon Q can involve a combination of the following API operations:

- **CreateApplication** – Creates an Amazon Q application
- **CreateWebExperience** – Creates an Amazon Q web experience
- **GetWebExperience** – Gets the properties of the web experience that you set up
- **ListWebExperiences** – Lists Amazon Q web experiences that you created
- **ChatSync** – Starts or continues a conversation in in your Amazon Q application

If you use the console to create your Amazon Q application, a web experience is created automatically and connected to your chosen data source. You can preview and deploy that web experience on the **Preview web experience** console page.

Before you can preview a web experience, make sure that you complete [creating your application](#).

**Topics**

- Preview and customize web experience
- Testing Amazon Q web experience functions
• Managing Amazon Q web experiences

Preview and customize web experience

The following tabs provide a procedure for previewing and customizing a web experience on the AWS Management Console and code examples for the AWS CLI.

Console

To preview and customize an Amazon Q web experience

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. Complete the steps to create your Amazon Q application.
3. Optionally, complete the steps to selecting an Amazon Q retriever, connecting data sources, and enhancing your application.
4. Then, from the Amazon Q application page, select Preview web experience.
5. In Preview web experience, from the right navigation pane, select Customize web experience.
6. In Customize web experience, enter the following information for your web experience:

   • Title – A title for your web experience. End users see this title on their web experience page.
   • Subtitle - optional – A subtitle for your web experience to highlight other information for your end users. This subtitle is visible to your end users on their web experience page.
   • Display welcome message – Provide an optional welcome message for your end users. We recommend mentioning data sources and application capabilities.
   • Display sample prompts – Provide a list of sample prompts on the end user’s conversation start screen.
7. Choose Save.
8. To exit the web experience preview and return to the Amazon Q console control panel to deploy your application, select Sign out from the left pane.

AWS CLI

To create and customize a web experience
Testing Amazon Q web experience functions

The following tabs provide a procedure for testing your web experience configuration for the AWS Management Console and code examples for the AWS CLI.

**Console**

**To test your Amazon Q web experience chat**

1. Sign in to the AWS Management Console and open the Amazon Q console at [https://console.aws.amazon.com/amazonq/](https://console.aws.amazon.com/amazonq/).
2. Complete the steps to [create your Amazon Q application](#).
3. Optionally, complete the steps to [selecting an Amazon Q retriever, connecting data sources](#), and [enhancing your application](#).
4. Then, from the Amazon Q application page, select **Preview web experience**.
5. Choose from the following options to test your web experience:

   a. **Ask questions** – Ask a question. Amazon Q generates and returns answers based on the enterprise data that the end user has access to. Continue the conversation by asking follow-up questions.

   b. **Verify response sources** – Each Amazon Q answer cites the source documents used to generate it.

   c. **See conversation history** – Amazon Q retains conversation history for 30 days so that they can search through questions and answers. You can view conversation history from the left navigation pane.

   d. **Summarize content** – Amazon Q can summarize email message threads.

   e. **Create outlines and drafts** – Use Amazon Q to create outlines and templates for documents.
f. **Perform plugin actions** – If you've configured [Plugins](#), ask Amazon Q to perform actions on your behalf, like creating a ticket in a supported third party app.

g. **Test guardrails and chat controls** – If you've configured [Guardrails and chat controls](#), check how Amazon Q responds to queries and special topics.

6. To exit the web experience preview and return to the Amazon Q console control panel to deploy your application, select **Sign out** from the left pane.

AWS CLI

**To preview web experience**

```plaintext
aws qbusiness chat-sync \
--application-id application-id \
--user-id user-id \
--user-groups optional-user-groups \
--user-message optional-user-message \
--action-execution optional-plugin-actions \
--attachments optional-file-uploads \
--attribute-filter optional-attribute-filters
```

Managing Amazon Q web experiences

To manage Amazon Q web experiences, you can take the following actions:

**Actions**

- [Creating a web experience](#)
- [Deleting a web experience](#)
- [Getting properties of a web experience](#)
- [Listing web experiences](#)
- [Updating a web experience](#)

**Creating a web experience**

To create an Amazon Q web experience, you can use the console or the CreateWebExperience API operation.
The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

Console

If you use the console, this action is spread across three steps: Configuring an Amazon Q application, Preview and customize web experience, and Deploying an Amazon Q web experience. Amazon Q creates a web experience for you when you configure your application. To create a web experience, you must create an application.

AWS CLI

To create an Amazon Q web experience

```bash
aws qbusiness create-web-experience \
--application-id application-id \
--sample-prompts-control-mode optional-sample-prompts \
--subtitle optional-subtitle \
--tags optional-tags \
--title optional-title \
--welcome-message optional-welcome-message \
```

Deleting a web experience

To delete an Amazon Q web experience, you can use the console or the DeleteWebExperience API operation.

If you're using the API, you can delete a web experience without deleting the application that it's a part of.

If you're using the console, the only way to delete your Amazon Q web experience is to delete the Amazon Q application that it's attached to.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

Console

To delete an Amazon Q web experience
1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.

2. In Applications, choose Actions.

3. Choose Delete.

4. In the dialog box that opens, type Delete to confirm deletion, and then choose Delete.

   You are returned to the service console while your application is deleted. When the deletion process is complete, the console displays a message confirming successful deletion. Both the application and the web experience are deleted.

AWS CLI

To delete an Amazon Q web experience

```bash
aws qbusiness delete-web-experience \  
   --application-id application-id \  
   --web-experience-id web-experience-id
```

Getting properties of a web experience

To get the properties of an Amazon Q web experience, you can use the console or the GetWebExperience API operation.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

Console

To get properties of an Amazon Q web experience

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.

2. In Applications, select the name of your application from the list of applications.

3. For Web experience settings, the following settings are available:

   - **Web experience IAM role ARN** – The IAM role assumed by end users when they log in to your web experience.
- **Deployed URL** – The deployed URL of your web experience.
- **Tags** – Tags that are attached to your web experience.

To update a setting, choose **Edit**.

### AWS CLI

#### To get properties of an Amazon Q web experience

```bash
aws qbusiness get-web-experience \
--application-id application-id \
--web-experience-id web-experience-id
```

### Listing web experiences

To list Amazon Q web experiences, you can use the console or the [ListWebExperiences](https://docs.aws.amazon.com/qbusiness/latest/APIReference/API_ListWebExperiences.html) API operation.

If you use the console, you can only see the web experience that's attached to a single application.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

#### Console

**To list Amazon Q web experiences**

1. Sign in to the AWS Management Console and open the Amazon Q console at [https://console.aws.amazon.com/amazonq/](https://console.aws.amazon.com/amazonq/).
2. For **Applications**, the Amazon Q web experience attached to your application is shown.

#### AWS CLI

**To list Amazon Q web experiences**

```bash
aws qbusiness get-web-experience \
```
Updating a web experience

To update an Amazon Q web experience, you can use the console or the UpdateWebExperience API operation.

The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

Console

To update an Amazon Q web experience

1. Sign in to the AWS Management Console and open the Amazon Q console at https://console.aws.amazon.com/amazonq/.
2. In Applications, select the name of your application from the list of applications.
3. On your application page, select Web experience settings and then select Edit.
4. On the Deploy web experience page, you can edit your web experience settings.

AWS CLI

To update an Amazon Q web experience

```bash
aws qbusiness update-web-experience \
--application-id application-id \
--web-experience-id web-experience-id \
--authentication-configuration optional-authentication-configuration \
--sample-prompts-control-mode optional-sample-prompts \
--subtitle optional-subtitle \
--title optional-title \
--welcome-message optional-welcome-message
```
Deploying an Amazon Q web experience

As the final step towards providing an Amazon Q web experience chat interface for your end users, you deploy the web experience that you created. Before you can deploy the web experience, you must set up end user authentication.

For your end users to log in and chat, Amazon Q requires that you integrate your web experience with an identity provider (IdP) that's compliant with SAML 2.0. This integration is required so that only authorized end users from within your organization have access to your content. Amazon Q can work with any IdP that's compliant with SAML 2.0. Amazon Q uses service-initiated single sign-on (SSO) to authenticate users. IdP-initiated SSO is not supported.

Note
During Preview, an Amazon Q application supports only 50 end users. If you need more capacity, contact Support.

To create and deploy your Amazon Q web experience, you can use either the AWS Management Console or the Amazon Q API. If you choose the API, use the CreateWebExperience API operation to create and deploy your web experience. Then, provide the deployment configuration information using the WebExperienceAuthConfiguration object.

If you use the console to create your Amazon Q application, a web experience is created automatically. Then, you deploy the web experience by specifying your configuration information on the console. If you use the console, setting up this connection involves copying and entering information from the Amazon Q console into the IdP console, and the other way around.

Topics

• Overview of integrating Amazon Q with an IdP
• Key concepts in integrating Amazon Q with an identity provider
• Steps for deploying your Amazon Q web experience
• Integrating Amazon Q with an identity provider
Overview of integrating Amazon Q with an IdP

To deploy an Amazon Q web experience, you must set up end user authentication. Amazon Q requires that you integrate your web experience with an identity provider (IdP) that's compliant with SAML 2.0. This integration is required so that only authorized end users from within your organization have access to your content. For more information, see Deploying an Amazon Q web experience.

The following gives you a high-level overview of the required steps to integrate Amazon Q with your IdP:

1. Create a new app integration in your IdP.
2. Share your Amazon Q configuration information with your IdP. This step starts the IdP and Amazon Q connection configuration process.
3. Share your IdP's federation metadata with Amazon Q. This step establishes a trust relationship between your IdP and Amazon Q. The trust relationship allows Amazon Q to validate user information that's communicated by your IdP. Establishing this trust relationship ensures that only a user who has permissions to access your application can access it.
4. Share the email attribute name (required) and group attribute name (optional) from your IdP with Amazon Q. Amazon Q uses this information to perform document access control based on the user's identity. This step ensures that your authenticated end user only sees chat responses generated from documents they have access to.

For more information about the terms used in describing the integration process, see Key concepts in integrating Amazon Q with an identity provider.

Key concepts in integrating Amazon Q with an identity provider

Amazon Q requires that you integrate your web experience with an identity provider (IdP) that's compliant with SAML 2.0. This integration is required so that only authorized end users from within your organization have access to your content. For more information, see Deploying an Amazon Q web experience. The following are key concepts that will help you understand the terms you encounter during the integration process:

Authorization

Authorization allows a user permissions to access specific resources.
Authentication

Authentication confirms a user's identity—that users are who they say they are.

Identity provider (IdP)

An identity provider (IdP) is a service that stores, manages, maintains, and verifies user identities for your application (in this case, Amazon Q). Some examples of IdPs are AWS IAM Identity Center, Okta, and Microsoft EntraID.

Service provider (SP)

A service provider (SP) is any entity—in this case, Amazon Q—that requests user authentication and authorization services from an IdP. Amazon Q takes the authentication information received from an IdP and uses it to authorize the end user's web experience session based on user authorization levels.

Security Assertion Markup Language (SAML)

SAML is an XML-based standard for transferring user identity data between the service provider (SP)—in this case, Amazon Q—and an identity provider (IdP) such as Okta, Ping, or EntraID. SAML supports two types of sign-in flows: Service initiated and IdP initiated.

Amazon Q only supports IdPs that are compliant with SAML 2.0.

Service provider-initiated single sign-on (SSO) flow

A SAML flow in which a service provider (SP) initiates the sign-in process.

⚠️ Important

Amazon Q uses service-initiated single sign-on (SSO) to authenticate users. IdP-initiated SSO is not supported.

Identity provider-initiated single sign-on (SSO) flow

A SAML flow in which the identity provider (IdP) (for example, Okta) initiates the sign-in process.

⚠️ Important

Amazon Q doesn't support IdP-initiated SSO.
Assertion consumer service (ACS) URL

An assertion consumer service (ACS) URL is an endpoint on the service provider (SP)—in this case, Amazon Q—where the IdP redirects its authentication response. This endpoint decides where your IdP sends its SAML response after authenticating a user.

Audience URI (SP entity ID)

The audience URI (service provider entity ID) is the unique ID of your service provider (SP). An identity provider (IdP) uses the audience URI to identify and direct its SAML response to a service provider.

XML metadata file

The XML metadata file is the document that contains the configuration information generated by your IdP during your SP-initiated single sign-on (SSO) process. The document contains the information needed for your SP and your IdP to trust and communicate with each other.

SAML assertion

A SAML assertion is a message that’s exchanged between your SP and your IdP that confidentially identifies a user. Assertions contain information about user identity, their group membership, the information that users can access, and any other relevant information.

Email attribute of SAML assertion

The email attribute of a SAML assertion is the attribute that your IdP maps user email to. For example, a user email address of mary_major@example.com can be mapped to the attribute user_email. Amazon Q uses this attribute value to resolve user access level to documents.

User group attribute of SAML assertion

The user group attribute of a SAML assertion is the attribute that the IdP maps user groups to. For example, the user groups "Research" and "Science" can be mapped to the attribute user_group. Amazon Q uses this attribute value to resolve user access level to documents.

Steps for deploying your Amazon Q web experience

This section guides you through the process of deploying your web experience by using the AWS Management Console console, the AWS Command Line Interface (AWS CLI), and the Amazon Q API. As a prerequisite, make sure you completed creating your application.
The following tabs provide a procedure for the AWS Management Console and code examples for the AWS CLI.

**Console**

**To deploy your Amazon Q web experience**

1. Sign in to the AWS Management Console and open the Amazon Q console at [https://console.aws.amazon.com/amazonq/](https://console.aws.amazon.com/amazonq/).
2. Complete the steps to [create your Amazon Q application](#).
3. Complete the steps for [selecting an Amazon Q retriever](#).
4. Complete the steps for [connecting data sources](#).
5. Optionally, complete the steps for [enhancing an application](#).
6. Optionally, complete the steps to [customize your web experience](#).
7. Then, in **Applications**, select your application, and choose **Deploy web experience**.
8. In **Service access**, enter the following information:

   - **Service access** – A service access role assumed by end users when they sign in to your web experience that grants them permission to start and manage conversations Amazon Q. You can choose to use an existing role or create a new role.
   - **Service role name** – A name for the service role you created for easy identification on the console.

9. From **Identity provider**, copy the following information to provide to the IdP you’re using:

   - **Assertion consumer service (ACS) URL** – Copy the ACS URL and enter it in the relevant section of your IdP.
   - **Audience URI (SP Entity ID)** – Copy the Audience URI (SP Entity ID) and enter it in the relevant section of your IdP.

10. In **Provide metadata from your IdP**, enter the following information:

    - Upload the metadata generated by your IdP as an XML file using **Import from XML**.

      See [Key IdP integration concepts](#) and [Integration process overview](#) for more details.

11. In **Configure user and group mapping**, enter the following information to allow ACLs to be active for end users using the web experience:

    - **Email attribute of SAML assertion** – Provide the attribute name that maps to user email.
• **User group field attribute of SAML assertion - optional** – Provide the attribute name that maps to user groups.

See [Key IdP integration concepts](#) and [Integration process overview](#) for more details.

12. To finish deploying your web experience, choose **Deploy**.

You are redirected to the Amazon Q control panel while your web experience deployment process finishes. After your application is deployed, your end users can access and chat in the web experience using the deployed web experience URL that's generated in the web experience details page by Amazon Q.

**AWS CLI**

**To deploy a web experience**

```bash
aws qbusiness create-web-experience \
--application-id application-id \
--metadata-xml metadata-xml \
--role-arn roleArn \
--user-id-attribute user-id-attribute \
--user-group-attribute user-group-attribute
```

**Integrating Amazon Q with an identity provider**

To deploy your Amazon Q web experience to your end users, you must integrate your Amazon Q application with an identity provider (IdP) that's compliant with SAML 2.0. You do this during the **deploy your web experience** process.

**Important**

Amazon Q uses service-initiated single sign-on (SSO) to authenticate users. IdP-initiated SSO is not supported.

This section walks you through the steps for integrating your Amazon Q application with SAML 2.0-compliant IdPs, such as AWS IAM Identity Center, Microsoft Entra ID (formerly known as Azure Identity Services).
Active Directory), and Okta. You can use similar steps for integrating your Amazon Q application with any IdP that's compliant with SAML 2.0.

To integrate your external SAML 2.0-compliant IdP, you must switch between tasks on the Amazon Q console and your IdP account.

For more information, see the following resources:

- For an overview of the Amazon Q web experience deployment process, see Process overview.
- For an overview of concepts related to the integration process, see Key concepts.

**Topics**
- Setting up Amazon Q with IAM Identity Center as identity provider
- Setting up Amazon Q with Microsoft Entra ID as identity provider
- Setting up Amazon Q with Okta as identity provider
- Setting up Amazon Q with PingIdentity as identity provider
- Troubleshooting Amazon Q and identity provider integration

**Setting up Amazon Q with IAM Identity Center as identity provider**

The following steps show how to set up Amazon Q with AWS IAM Identity Center as your SAML 2.0-compliant identity provider (IdP). Integrating Amazon Q with IAM Identity Center requires that you switch between tasks on the Amazon Q console and the IAM Identity Center console.

**Prerequisites**

Before you start to integrate Amazon Q with IAM Identity Center, make sure that you have completed the following tasks:

- Created an Amazon Q application, selected a retriever, added your desired data sources, and previewed an Amazon Q web experience.
- Enabled an IAM Identity Center instance, provisioned at least one user, and provided each user with a valid email address.

**To integrate Amazon Q with IAM Identity Center**

1. In the Amazon Q console, choose your application for integrating with IAM Identity Center.
2. On the **Application** page, scroll down and choose the **Web experience settings** tab. Choose **Edit**.

3. For **Service role name**, choose the IAM role that you created for your web experience. Or, choose **Create a new role**.

4. In the **Configure your identity provider** section, copy **Assertion consumer service (ACS) URL** and **Audience URI (SP Entity ID)**. You will use them later in this procedure.

5. Go to the IAM Identity Center console. In the left navigation pane, expand **Application assignments** and choose **Applications**.

6. On the **Applications** page, choose **Add application**.

7. On the **Select application type** page, for the **Setup preference**, choose **I have an application I want to set up**.

8. In the **Application type** section, choose **SAML 2.0** and choose **Next**.

9. On the **Configure application** page, enter a name for your application. Optionally, enter a description.

10. In the **User and group assignment method** section, choose **Require assignments**.

11. In the **IAM Identity Center metadata** section, choose **Download** to download the IAM Identity Center SAML metadata file.

12. Scroll down to the **Application metadata** section, and choose **Manually type your metadata values**.

13. Paste the corresponding **Assertion Consumer Service (ACS) URL** and **Application SAML audience URI** that you copied from the IAM Identity Center console. Choose **Submit**.
14. On the **Custom SAML 2.0 application** page, in the **Assigned users** section, choose **Assign users**.

15. In the **Users** table, select one or more users for your application and then choose **Assign users** to finish assigning users.

16. In the **Details** pane, choose **Actions** and then choose **Edit attribute mappings**.
17. On the **Attribute mappings** page, do the following:

   a. Leave the **User attribute in the application** column set to the default attribute name **Subject**.

   b. For **Maps to this string value or user attribute in IAM Identity Center**, map the Subject to the email attribute, for example, `${user:email}`. Make sure that the attribute you provide is included in [Supported IAM Identity Center attributes](#).

18. Set the **Format** to **unspecified**.

19. Choose **Add new attribute mapping**.

   a. For **User attribute in the application**, enter a name for the attribute, for example, Email. Make a note of this attribute name for use later.

   b. For **Maps to this string value or user attribute in IAM Identity Center**, enter an attribute or a value that you want to map to the attribute name. For example, you might want to map the attribute name Email with the users email attribute `${user:email}`. Make sure that the attribute you provide is included in [Supported IAM Identity Center attributes](#).

   c. Set the **Format** to **unspecified**.

20. Choose **Save changes**.
21. Go back to the Amazon Q console, and make sure you're on the **Deploy web experience** page.

22. Scroll down to the **Provide metadata from your IdP** section. To upload the metadata XML file that you saved in your previous steps, choose **Import from XML**.

23. In the **Configure user and user group** section, for **Email attribute of SAML assertion**, enter the attribute name that you provided in the IAM Identity Center console. For example, **Email** could be an attribute name. Choose **Deploy**.

24. On your Amazon Q application page, a URL should appear under **Deployed URL**.

25. Choose the URL to open your Amazon Q web experience and enter credentials for a user that has access to the web experience.

    If you encounter HTTP status code 403 (Forbidden) errors, see [Troubleshooting Amazon Q and identity provider integration](#).

### Setting up Amazon Q with Microsoft Entra ID as identity provider

The following steps show how to set up Amazon Q with Microsoft Entra ID (formerly Azure Active Directory) as your SAML 2.0-compliant identity provider. Integrating Amazon Q with Entra ID requires that you switch between tasks on the Amazon Q console and in the Entra ID portal.

#### Prerequisites

Before you start to integrate Amazon Q with Entra ID, make sure that you have completed the following tasks:

- Created an Amazon Q application, selected a retriever, added your desired data sources, and previewed Amazon Q web experience.
- Created an Entra ID instance, provisioned at least one user, and provided each user with a valid email address.
To integrate Amazon Q with Entra ID

1. In the Amazon Q console, choose your application for integrating with Microsoft Entra ID.
2. On the Application page, scroll down and choose the Web experience settings tab. Choose Edit.
3. For Service role name, choose the IAM role that you created for your web experience. Or, choose Create a new role.
4. In the Configure your identity provider section, copy Assertion Consumer Service (ACS) URL and Audience URI (SP Entity ID). You will use them later in this procedure.
5. Go to the Entra ID portal. In the left navigation pane, choose Enterprise applications, and then choose Add.
6. On the All applications page, choose New application.
7. In the Browse Microsoft Entra Gallery page, choose Create your own application.
8. Enter a name for your application, choose **Integrate any other application you don't find in the gallery (Non gallery)**, and choose **Create**. It might take a few minutes for your application to be provisioned.

**Create your own application**

Got feedback?

If you are developing your own application, using Application Proxy, or want to integrate an application that is not in the gallery, you can create your own application here.

What’s the name of your app?

Input name

What are you looking to do with your application?

- Configure Application Proxy for secure remote access to an on-premises application
- Register an application to integrate with Microsoft Entra ID (App you’re developing)
- Integrate any other application you don't find in the gallery (Non-gallery)

Create

9. On the **Application overview** page, in the **Getting started** section, choose **Set up single sign on**.

10. In the **Select a single sign-on method** pane, choose **SAML**.
11. In the **Basic SAML Configuration** section, choose **More** (three dots) and then choose **Edit**.

12. Choose **Add identifier**. For the **Identifier (Entity ID)** field, enter the **Audience URI (SP Entity ID)** that you copied from the Amazon Q console. Next, choose **Add rely URL**. For the **Reply URL (Assertion Consumer Service URL)** field, enter the **Application consumer service (ACS) URL** that you copied from the Amazon Q console.

13. Leave the rest of the fields blank. Choose **Save**.
14. On the **Set up single sign-on with SAML** page, scroll down to the **SAML Certificates** section. Download the **Federation Metadata XML** file and save it in your local drive.
15. In the **Attributes & Claims** section, choose **More** (three dots) and then choose **Edit**.

---

**Basic SAML Configuration**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifier (Entity ID)</td>
<td>Required</td>
</tr>
<tr>
<td>Reply URL (Assertion Consumer Service URL)</td>
<td>Required</td>
</tr>
<tr>
<td>Sign on URL</td>
<td>Optional</td>
</tr>
<tr>
<td>Relay State (Optional)</td>
<td>Optional</td>
</tr>
<tr>
<td>Logout URL (Optional)</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**Attributes & Claims**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>givenname</td>
<td>user.givenname</td>
</tr>
<tr>
<td>surname</td>
<td>usersurname</td>
</tr>
<tr>
<td>emailaddress</td>
<td>user.mail</td>
</tr>
<tr>
<td>name</td>
<td>user.userprincipalname</td>
</tr>
<tr>
<td>Unique User Identifier</td>
<td>user.userprincipalname</td>
</tr>
</tbody>
</table>

**SAML Certificates**

<table>
<thead>
<tr>
<th>Certificate</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token signing certificate</td>
<td>Active</td>
</tr>
<tr>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>Thumbprint</td>
<td></td>
</tr>
<tr>
<td>Expiration</td>
<td>8/28/2023, 1:02:40 PM</td>
</tr>
<tr>
<td>Notification Email</td>
<td></td>
</tr>
<tr>
<td>App Federation Metadata Url</td>
<td><a href="https://login.microsoftonline.com/888d0b57-69f1...">Download</a></td>
</tr>
<tr>
<td>Certificate (Base64)</td>
<td>Download</td>
</tr>
<tr>
<td>Certificate (Raw)</td>
<td>Download</td>
</tr>
<tr>
<td>Federation Metadata XML</td>
<td>Download</td>
</tr>
</tbody>
</table>

**Verification certificates (optional)**

<table>
<thead>
<tr>
<th>Certificate</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td>No</td>
</tr>
<tr>
<td>Active</td>
<td>0</td>
</tr>
<tr>
<td>Expired</td>
<td>0</td>
</tr>
</tbody>
</table>
Set up Single Sign-On with SAML

An SSO implementation based on federation protocols improves security, reliability, and end user experiences and is easier to implement. Choose SAML single sign-on whenever possible for existing applications that do not use OpenID Connect or OAuth. Learn more.

Read the configuration guide for help integrating.

In the Attributes & Claims page, choose Unique User Identifier (Name ID).
17. In the **Manage claim** page, expand **Choose name identifier format**. For the **Name identifier format** field, select **Unspecified**. Choose **Save**.

18. In the **Attributes & Claims** page, choose **Add new claim**.
19. For the **Name** field, enter **Email**.

20. Expand **Choose name format**.
   
   a. For the **Name format** field, select **Unspecified**.
   
   b. Make sure that the **Source** is set to **Attribute**.
   
   c. For the **Source attribute** field, choose the drop-down arrow and select **user.mail**.
   
   d. Choose **Save**.

21. Go back to your application page. In the left navigation pane of your application page, choose **Users and groups**.

22. In the **Users** table, select the user that you created earlier. To finish assigning users, choose **Assign**. Continue with the next steps.
a. If you do not see the user you want to add to your application, choose + Add user/group.

b. In the Add Assignment page, choose None Selected.

c. In the right pane, select the user or search for the user in the search bar and then select the user.

d. Choose Select and then choose Assign.

23. In the Users and groups page, choose the user name. On the user page, verify that the User principal name and Identities fields are populated.

24. Go back to the Amazon Q console. Scroll down to Provide metadata from your IdP.

25. Select Import from XML file to upload the metadata XML file that you downloaded from the Entra ID portal.

26. In the Configure user and user group section, for Email attribute of SAML assertion, enter the attribute name that you provided in the Entra ID portal. For example, Email could be an attribute name. Choose Deploy.

27. On your Amazon Q application page, a URL should appear under Deployed URL.

28. Select the URL to open your Amazon Q web experience and enter the credentials for a user that has access to the web experience.

If you encounter or any other issues, see Troubleshooting Amazon Q and identity provider integration.

Setting up Amazon Q with Okta as identity provider

The following steps show how to integrate Amazon Q with Okta as your SAML 2.0-compliant identity provider (IdP). Integrating Amazon Q with Okta requires that you switch between tasks on the Amazon Q console and the Okta admin console.

Prerequisites

Before you start to integrate Amazon Q with Okta, make sure that you have completed the following tasks:

• Created an Amazon Q application, selected a retriever, added your desired data sources, and previewed Amazon Q web experience.
• Created an Okta account, added at least one user, assigned users to their groups, and provided each user with a valid email address. For more information, see Manage users on the Okta Help Center.

To integrate Amazon Q with Okta

1. In the Amazon Q console, choose your application for integrating with Okta.
2. On your application page, scroll down and choose the Web experience settings tab. Choose Edit.
3. On the Deploy web experience page, for Service role name, choose the IAM role that you created for your web experience. Or, choose Create a new role.
4. In the Configure your identity provider(IdP) section, copy Assertion Consumer Service (ACS) URL and Audience URI (SP Entity ID). You will use them later in this procedure.
5. Go to the Okta admin console. In the left navigation pane, choose Applications, and then choose Create App Integration.

6. On the Create a new app integration page, choose SAML 2.0 and then choose Next.
7. On the **Create SAML Integration** page, for **General Settings**, in **App name**, enter a name for the application and choose **Next**.

8. For **Configure SAML**, in the **SAML Settings** section, do the following:

   a. For the **Single sign-on URL** field, enter the **Assertion Consumer Service (ACS) URL** that you copied from the Amazon Q console.
b. For the **Audience URI (SP Entity ID)** field, enter the **Audience URI (SP Entity ID)** that you copied from the Amazon Q console.

9. Scroll down to the **Attribute Statements (optional)** section, and provide the following information. This information will be used by the Amazon Q application to identify the end user's email address.

   a. For the **Name** field, provide a name for the email attribute, for example **Email**.
b. For the **Name format** field, leave it set to **Unspecified**.

c. For the **Value** field, provide a mapping to the attribute by selecting `user.email` from the dropdown list.

d. (Optional) To add users, choose **Add another** and provide an attribute name and a value for each user. Make sure to leave the name format set to **Unspecified** for each user.

e. Choose **Next**, and then choose **Finish**.

10. From your Okta app **Settings** page, select the **Assignments** tab.

11. Select **Assign**. To assign users to your Okta app, choose between **Assign to People** and **Assign to Groups**.

12. To finish assigning users, choose **Done**.

13. Go back to the Okta app **Settings** page, and select the **Sign-on** tab.

14. In the **Metadata details** section, to copy the metadata file XML file and save it in `.xml` format, choose **Copy**.
**Sign on methods**

The sign-on method determines how a user signs into and manages their credentials for an application. Some sign-on methods require additional configuration in the 3<sup>rd</sup> party application.

Application username is determined by the user profile mapping. Configure profile mapping

---

**SAML 2.0**

**Default Relay State**

**Metadata details**

| Metadata URL | https://trial-8515555.okta.com/app/exk60mng/sw71XUb697/ss0/saml/metadata |

More details
Note

You can also navigate to the metadata URL and copy the network response payload and paste it in a file that you save in .xml format.

For more information, see Create SAML app integrations on the Okta Help Center website.

15. Return to the Amazon Q console and make sure that you are on the Deploy web experience page.
16. Scroll down to the Provide metadata from your IdP section. Select Import from XML file to upload the metadata XML file you copied from the Okta console.
17. In the Configure user and user group section, for Email attribute of SAML assertion, enter the attribute name that you provided for your Okta user. For example, Email could be an attribute name. Choose Deploy.
18. On your Amazon Q application page, a URL should appear under Deployed URL.
19. Choose the URL to open your Amazon Q web experience and enter the credentials for a user that has access to the web experience.

If you encounter HTTP status code 403 (Forbidden) errors or any other issues, see Troubleshooting Amazon Q and identity provider integration

Setting up Amazon Q with PingIdentity as identity provider

The following steps show how to integrate Amazon Q with PingIdentity (Ping) as your SAML 2.0-compliant identity provider (IdP). Integrating Amazon Q with Ping requires that you switch between tasks on the Amazon Q console and your PingIdentity console.

Prerequisites

Before you start to integrate Amazon Q with Ping, make sure that you have completed the following tasks:

- Created an Amazon Q application, selected a retriever, added your desired data sources, and previewed Amazon Q web experience.
- Created a PingIdentity account, added at least one user, and provided each user with a valid email address.
To integrate Amazon Q with Ping

1. In the Amazon Q console, choose your application for integrating with Ping.

2. In the Application page, scroll down and choose the Web experience settings tab. Choose Edit.

3. For Service role name, choose the IAM role that you created for your web experience. Or, choose Create a new role of your Amazon Q application.

4. In the Configure your identity provider section, copy the Assertion Consumer Service (ACS) URL and the Audience URI (SP Entity ID). You will use them later in this procedure.

5. Go to the PingIdentity console. In the left navigation pane, choose Applications.

6. Choose the plus sign (+) next to Applications to create a new application.

7. In the Add Application section, enter a name for your application and optionally enter a description.
8. In the **Application Type** section, choose **SAML Application** and then choose **Configure**.

9. In the **SAML Configuration** section, choose **Manually Enter** and then do the following:

   a. For **ACS URLs**, paste the **Application consumer service (ACS) URL** that you copied from the Amazon Q console.

   b. For **Entity ID**, paste the **Audience URI (SP Identity)** that you copied from the Amazon Q console.
10. Choose **Save**.

11. In your application page, choose **Configuration** and then choose **Edit**.

12. Scroll down to the **SUBJECT NAMEID FORMAT** field, set the format to **unspecified**, and then choose **Save**.
The format name will look similar to `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`.

**Encryption**
- **Enable Encryption**

**Entity ID**

**SLO Endpoint**

**Subject NameID Format**
- `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`

**Assertion Validity Duration (in seconds)**
- 300

**Target Application URL**

- **Enforce Signed AuthnRequest**

**Verification Certificate (Optional)**
- **None**
- **Import**
- **Choose from list**

- **Select Policy based on RequestedAuthnContext**

**Save**

**Cancel**
13. On your application page, choose **Attribute Mappings** and then choose **Edit**.

14. On the **Attribute Mapping** page, provide the following information for your application to identify the end user's email address:

   a. For the **saml_subject** attribute, leave the **PingOne Mappings** set to **User ID**.

   b. Choose the update button (three vertical dots), choose **Update NameFormat**, and set the name format to **unspecified**.

   The format name will look similar to `urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified`.

   c. Choose **Update** and then choose **Add**.
d. Enter a name for the email attribute, for example, **Email**.

e. Set the **PingOne Mappings** for email attribute to **Email Address**.

f. Choose **Save**.

15. Choose **Configuration**. On the **Connection Details** page, choose **Download Metadata**.

16. Choose the enable button next to your application name to enable your application.
17. Go back to the Amazon Q console, and scroll down to **Provide metadata from your IdP**. To upload the metadata XML file that you downloaded from the PingIdentity console, choose **Import from XML file**.

18. In the **Configure user and user group** section, for **Email attribute of SAML assertion**, enter the attribute name that you provided in the PingIdentity console. For example, **Email** could be an attribute name. Choose **Deploy**.

19. On your Amazon Q application page, an URL should appear under **Deployed URL**.

20. Choose the URL to open your Amazon Q web experience and enter the credentials for a user that has access to the web experience.

If you encounter HTTP status code 403 (Forbidden) errors or any other issues, see [Troubleshooting Amazon Q and identity provider integration](#).

### Troubleshooting Amazon Q and identity provider integration

This topic helps you troubleshoot issues with opening an Amazon Q application after you have integrated Amazon Q with an identity provider.

If you encounter an HTTP status code 403 (Forbidden) error when you open your Amazon Q application, try the following troubleshooting steps:

- Check the attribute mappings in your identity provider's console. Make sure that the subject attributes and email attributes are set to the **unspecified** format.
• Check that the name you entered in the Amazon Q console for **Email attribute** matches the name that you specified in your identity provider attribute mappings page.

• Verify that the user you used to sign in with has access to the web experience. Check the **Assignments** section on your identity provider application page, and confirm that the user is listed and assigned to the web experience.

• Verify that the user you used to sign in with has a value defined for their email address. Verify that this value is correctly mapped to the email attribute mapping that you configured.
Amazon Q features

In addition to enhancements, Amazon Q offers the following features:

- **Filtering using metadata** – Use document attributes to customize and control the end user chat experience. Currently supported only if you use the Amazon Q API.
- **Source attribution with citations** – Verify responses using Amazon Q source attributions.
- **Upload files and chat** – Let end users upload files directly into chat and use uploaded file data to perform web experience tasks.
- **Quick prompts** – Feature sample prompts to inform end users of the capabilities of their Amazon Q web experience.

Topics

- **Filtering using document attributes in Amazon Q**
- **Source attribution with citations in Amazon Q**
- **Upload files and chat in Amazon Q**
- **Quick prompts in Amazon Q**

Filtering using document attributes in Amazon Q

If you use the API, Amazon Q includes a filtering by document attribute feature. With this feature, you can customize and control chat responses for your end user using attributes—or metadata attached to documents. For example, if data source type is an attribute attached to your documents, you can specify that chat responses be generated only from a specific data source like Slack.

Or, you can allow end users to restrict the scope of chat responses using the attribute filters that you have selected. For example, an end user can choose that their chat responses be generated using documents from specific data sources.

Filtering chat responses using metadata has the following key benefits:

- **Ensure response relevance and accuracy** – You can specify that responses be generated from and limited to authoritative sources within your data
• **Control response context** – You can specify the type (PDF, for example) and corpus (Business Requirement Documents, for example) of documents that responses will be generated from.

• **Maintain response freshness** – You can restrict chat responses to only documents that were generated after a specific date.

• **Scope chat responses** – You can help your end user narrow the scope of their responses and get to the right answer quicker.

Amazon Q offers a set of reserved document attributes that you can use. You can also create custom document attributes that are more representative of your organization’s data and use cases for more fine-grained chat response control.

⚠️ **Important**
Currently, the filtering using document attributes feature is only supported through the API.

**Topics**
- Understanding document attributes
- Mapping document attributes

**Understanding document attributes**

Every document has structural attributes—or metadata—attached to it. Document attributes can include information such as document title, document author, time created, time updated, and document type.

You can map document attributes to fields in your Amazon Q index. Once mapped to document attributes, these index fields can be used by end users to filter and scope their chat results to specific data.

⚠️ **Note**
Currently, filtering using document attributes in chat is only supported through the API.

**Topics**
Reserved document attributes

Amazon Q offers the following reserved or default document attributes or index fields that you can map your metadata to:

- **_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 25, 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.
- **_file_type** – The file type of the document, such as .pdf or .docx.
- **_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 25, 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.

Custom document attributes

You can also create custom attributes based on your own enterprise data. Then, you can map the custom attributes to custom index fields that you create for a more tailored end user chat experience.
For example, you can create a custom field or attribute called "Department" with the values of "HR", "Sales", and "Manufacturing". You can then use these fields or attributes to allow your end user to filter their chat results to documents in the "HR" department, or restrict response generation to specific data stores.

You can create up to 50 custom fields or attributes.

⚠️ Important

Once created, you can't delete or rename any attributes.

Topics

- Ingesting attributes using the BatchPutDocument API operation
- Adding custom attributes or fields to an Amazon S3 data source

Ingesting attributes using the BatchPutDocument API operation

When you use the BatchPutDocument operation to add a document to your index, you can specify document attributes—both reserved and custom—as part of Attributes. You can add multiple fields or attributes when you call the API operation. You can create up to 50 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 Amazon 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 50 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"]
}

**Mapped document attributes**

When a document attribute—reserved or custom—is mapped to an index field, you can configure it to perform the following action:

- **Search** – Allows end users the ability to search data with the specified attributes.

**Document attribute data types**

Document attributes—reserved or custom—can only be the data types that are shown in the following table.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Searchable</th>
<th>Filterable</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>No</td>
<td>Yes</td>
<td>DISABLED</td>
</tr>
<tr>
<td>Number</td>
<td>No</td>
<td>Yes</td>
<td>DISABLED</td>
</tr>
<tr>
<td>String</td>
<td>Yes</td>
<td>Yes</td>
<td>ENABLED</td>
</tr>
<tr>
<td>String list</td>
<td>Yes</td>
<td>Yes</td>
<td>ENABLED</td>
</tr>
</tbody>
</table>

**Note**

You can't change an index field type after it has been created.
Mapping document attributes

Mapping document attributes from your documents to index fields is a multi-step process that depends on the document upload method you use.

Note
Currently, filtering using document attributes in chat is only supported through the API.

Topics
- Mapping document attributes directly to index fields
- Mapping data source document attributes to index fields

Mapping document attributes directly to index fields

When you use the API, you must first map your document attributes to index fields before you can use them for filtering in chat. You use the following process to map document attributes to your index field:

1. You create an index by calling the CreateIndex API operation.
2. Then, you create index fields using the UpdateIndex operation. You use this method to map both reserved and custom document attributes to index fields.
3. Optionally, you can test and view the index fields that you've added by using the GetIndex operation.
4. Then, when you use the BatchPutDocument operation to ingest documents into your index, Amazon Q extracts your reserved or custom document attributes and maps them to the index fields that you have already created.

After you map document attributes directly to index fields using the API, you can select specific attributes for your end user to use for filtering chat responses. With the UpdateIndex API operation, you add custom fields or attributes using the documentAttributeConfigurations parameter.

The following JSON example uses documentAttributeConfigurations to add a field called "Department" to the index.
"DocumentmetadataConfigurationUpdates": [
  {
    "Name": "Department",
    "Type": "STRING_VALUE"
  }
]

Mapping data source document attributes to index fields

If you use an Amazon Q data source connector, you can map common or default document attributes attached to documents in your data source to fields in your Amazon Q index. You can use these document attributes to help your end user filter and scope chat responses.

⚠️ Important

Currently, filtering using data source document attributes in chat is only supported through the API.

Each data source connector is designed to crawl the default document attributes in your data source automatically. For example, if you have a field in your data source named dept that contains department information for a document, you can map it to an index field named Department. You can't change or customize default data source attributes that are mapped to an index.

You can also map any Amazon Q reserved or common fields such as _created_at. If your data source has a field named creation_date, you can map this field to the equivalent Amazon Q reserved field named _created_at.

You can also choose to add custom document attributes and map them to custom fields that you create in your index. Most data sources support field mappings and follow a specific configuration format, except Amazon S3 and database data sources:

- If you store your documents in an Amazon S3 bucket or Amazon S3 data source, you can either use the console to specify field mappings or specify fields using a JSON metadata file.
- For database data sources, if the name of the database column matches the name of a reserved field, the field and column are mapped automatically.
If you use the console, you select default field mappings or create custom mappings when you configure your connector. On the console, if a default field or a default field property can't be edited, it will appear grayed out.

**Note**
Currently, you can't use field mappings configured on the console to filter chat responses.

If you use the API, you use the configuration parameter of the [CreateDataSource](#) API operation to map default document attributes in your data source to index fields. If you want to map custom document attributes in your data source to Amazon Q index fields, use the DocumentAttribute parameter of the [UpdateIndex](#) operation to first create the custom field matching the custom document attribute. By doing so, you can specify and map your reserved or custom data source document attribute to a reserved or custom index field.

**Source attribution with citations in Amazon Q**

The Amazon Q web experience chat response provides in-text source citations for responses that use the organization's data sources and knowledge base as a source. The chat response also provides an entire list of sources used to generate the response.

**In-text source citations**

In-text citations are provided in the form of a numbered list at the end of a sentence. To view an in-text source citation, choose a citation number. Each citation provides the following attributes:

- **Title** – The title of the document that's the source for the generated response.
- **URL** – The URL of the document that's the source for the generated response. Choose the URL to view the source document.

**Source list**

Sources used to generate the response are provided at the end of the response. Each source listed provides the following attributes:

- **Citation number** – The number provided at the end of the sentences in the response.
- **Title** – The title of the document that's the source for the generated response.
Upload files and chat in Amazon Q

End users using the Amazon Q web experience can upload documents that might not be stored in your organization's data sources and knowledge base. They can use the uploaded documents to ask questions and summarize or analyze data that's based on the content of the uploaded documents. The uploaded documents aren't stored and are available for use only for the conversation in which the documents are uploaded.

You can upload up to 5 files during a conversation. The size of each file you upload must be 10 MB or less. The parsed content for each file has to be under 30,000 tokens or 20,000 words.

Amazon Q supports specific document types for upload. To learn more about the document types that can be uploaded, see Supported document formats in Amazon Q.

Quick prompts in Amazon Q

The Amazon Q web experience welcome page provides sample prompts to help end users understand the types of questions and tasks that they can ask in the web experience. Sample prompts aren't enabled by default.

If you're an AWS Management Console customer and are configuring the web experience for your end users, you can enable the sample prompts feature when you preview the web experience. For more information, see Preview and customize web experience.

⚠️ Important
Before you enable the sample prompts feature, make sure that the Only produce responses from retrieval augmented generation (RAG) check box for the Application guardrails is cleared. For more information, see Customizing global controls. The sample prompts might not work if the responses is restricted to enterprise data.

Currently, you can't create your own prompts or edit the provided sample prompts.
Using an Amazon Q web experience

An Amazon Q web experience is an easy-to-use generative artificial intelligence (generative AI) assistant. You can use the Amazon Q web experience to ask questions and to accomplish your tasks. When you ask a question, the Amazon Q web experience analyzes the latest approved data collected from various data sources within your organization to generate a comprehensive response.

With an Amazon Q web experience, you can ask complex questions in plain language and get a detailed response. You can also use an Amazon Q web experience to perform tasks for you, such as draft an email message or create a Jira ticket.

The Amazon Q web experience provides you with the following capabilities:

**Web experience features**

- Prompts
- Engage with contextual responses
- Analyze content
- Perform actions on your behalf
- Review source citations
- Upload files and chat
- Copy responses
- Provide feedback
- Conversation management

**Prompts**

The welcome page optionally provides example prompts to help you understand the types of questions and tasks that you can ask the Amazon Q web experience. This feature is provided depending on how the web experience is configured. If provided, use the sample prompts to formulate your own questions and tasks.
Engage with contextual responses

The Amazon Q web experience analyzes your questions and returns responses that use information from various data sources within your organization. You can continue with the conversation in the context of the active session or start a new conversation.

Analyze content

Ask the Amazon Q web experience to summarize its response, generate text from the response, do comparative analysis, and also perform math and reasoning tasks.

Perform actions on your behalf

Use the Amazon Q web experience to perform actions on your behalf. For example, you can ask the web experience to schedule a meeting, create a ticket in Jira, or draft an email message.

Review source citations

The Amazon Q web experience provides in-text source citations in the form of a numbered list. To view the source of the response, choose the number at the end of the sentence.

To view the entire list of sources, choose Sources at the end of the response. Use the source list to fact-check the response or for deeper analysis.

Upload files and chat

With the Amazon Q web experience, you can upload documents that aren't stored in your organization's data sources and knowledge base. Then you can use the uploaded documents to ask questions and summarize or analyze data that's based on the content of the uploaded documents. The uploaded documents aren't stored and are available for use only during the session in which the documents are uploaded.

To upload documents during a session, choose the upload icon next to the question box. You can upload a maximum of five files in a single session.
Copy responses

You can copy and save the responses for later review and analysis. To copy a response, choose the copy icon at the end of the response.

Provide feedback

To provide immediate feedback about the response you received from the Amazon Q web experience, use the thumbs-up or thumbs-down button. Your feedback is used to help address technical issues in the web experience.

Conversation management

Amazon Q web experience stores each conversation for up to 30 days. Your conversations are listed in the left navigation pane. You can perform the following tasks to manage your conversations:

- **View conversation history** – Choose a conversation to view the conversation history for that session.
- **Start new conversation** – Choose + New conversation to start a new conversation.
- **Delete conversation** – Choose a conversation that you want to delete, choose Delete, and then choose Delete again.
Tagging resources

Manage your Amazon Q applications and data sources by assigning tags. You can use tags to categorize your Amazon Q resources in various ways. For example, you could categorize 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 do the following:

- **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 Amazon Kendra retriever and the Amazon Q web experience that uses the retriever 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](https://docs.aws.amazon.com/billing/latest/userguide/Cost-Allocation-Tagging.html) in the *AWS Billing User Guide*.

- **Control access to your resources** – You can use tags in AWS Identity and Access Management (IAM) policies that control access to Amazon Q resources. To activate tag-based access control, you can attach these policies to an IAM role or IAM user. For more information, see [Authorization based on tags](https://docs.aws.amazon.com/IAM/latest/userguide/permission-tag-based-analysis.html).

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

**Topics**

- [Using tags](#)
- [Tag restrictions](#)

**Using tags**

If you're using the 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 CLI or the Amazon Q API, use the following operations to manage tags for your resources:
• **CreateApplication** – Apply tags when you create an Amazon Q application.
• **CreateDataSource** – Apply tags when you create a data source.
• **CreateIndex** – Apply tags when you create an Amazon Q retriever and index.
• **CreateRetriever** – Apply tags when you create an Amazon Kendra retriever.
• **CreateWebExperience** – Apply tags when you create an Amazon Q web experience.
• **CreatePlugin** – Apply tags when you create an Amazon Q plugin.
• **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 Q 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
Security in Amazon Q

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are 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 Q, 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 Q. The following topics show you how to configure Amazon Q 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 Q resources.

**Topics**

- Data protection in Amazon Q
- Identity and access management for Amazon Q
- Compliance validation for Amazon Q
- Resilience in Amazon Q
- Infrastructure security in Amazon Q
- Cross-service confused deputy prevention
- Configuration and vulnerability analysis in AWS Identity and Access Management
- Security best practices
Data protection in Amazon Q

The AWS shared responsibility model applies to data protection in Amazon Q. 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.
- 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 140-2 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) 140-2.

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 Q 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.

Topics

- Data encryption
- Key management
Data encryption

Amazon Q supports encryption at rest using an AWS KMS key that's owned by AWS. Amazon Q also uses HTTPS protocol for data in transit.

Topics

- Encryption at rest
- Encryption in transit

Encryption at rest

Amazon Q provides encryption by default to protect sensitive customer data at rest using AWS owned encryption keys.

- **AWS owned keys** – Amazon Q uses these keys by default to automatically encrypt personally identifiable data. You can't view, manage, or use AWS owned keys, or audit their use. However, you don't have to take any action or change any programs to protect the keys that encrypt your data. For more information, see [AWS owned keys](https://docs.aws.amazon.com/kms/latest/developerguide/aws-owned-keys.html) in the *AWS Key Management Service Developer Guide*.

Encryption of data at rest by default helps reduce the operational overhead and complexity involved in protecting sensitive data. At the same time, it enables you to build secure applications that meet strict encryption compliance and regulatory requirements.

While you can't disable this layer of encryption or select an alternate encryption type, you can add a second layer of encryption over the existing AWS owned encryption keys by choosing a customer managed key when you create your resources:

- **Customer managed keys (CMK)** – Amazon Q supports the use of a symmetric customer managed key that you create, own, and manage to add a second layer of encryption over the existing AWS owned encryption. Because you have full control of this layer of encryption, you can perform such tasks as:
  - Establishing and maintaining key policies
  - Establishing and maintaining IAM policies and grants
  - Enabling and disabling key policies
  - Rotating key cryptographic material
  - Adding tags
• Creating key aliases

• Scheduling keys for deletion

For more information, see customer managed key in the AWS Key Management Service Developer Guide.

Note
If you have created your Amazon Q application using AWS KMS and then you want to migrate to using customer managed key (CMK), you will have to re-create your application.

Topics
• How Amazon Q uses grants in AWS KMS
• Create a customer managed key (CMK)
• Specifying customer managed key for Amazon Q
• Monitoring your encryption keys for Amazon Q

How Amazon Q uses grants in AWS KMS

Amazon Q requires a grant to use your customer managed key. When you create a Amazon Q application resource encrypted with a customer managed key, Amazon Q creates a grant on your behalf by sending a CreateGrant request to AWS KMS. Grants in AWS KMS are used to give Amazon Q access to a KMS key in a customer account.

Amazon Q requires the grant to use your customer managed key for the following internal operations:

• Send DescribeKey requests to AWS KMS to verify that the symmetric customer managed key ID entered when creating application is valid.

• Send GenerateDataKeyWithoutPlainText requests to AWS KMS to generate data keys encrypted by your customer managed key.

• Send Decrypt requests to AWS KMS to decrypt the encrypted data keys so that they can be used to encrypt your data.
You can revoke access to the grant, or remove the service's access to the customer managed key at any time. If you do, Amazon Q won't be able to access any of the data encrypted by the customer managed key, which affects operations that are dependent on that data.

**Create a customer managed key (CMK)**

You can create a symmetric customer managed key by using the AWS Management Console, or the AWS KMS APIs.

**To create a symmetric customer managed key**

Follow the steps for Creating symmetric customer managed key in the AWS Key Management Service Developer Guide.

**Key policy**

Key policies control access to your customer managed key. Every customer managed key must have exactly one key policy, which contains statements that determine who can use the key and how they can use it. When you create your customer managed key, you can specify a key policy. For more information, see Managing access to customer managed keys in the AWS Key Management Service Developer Guide.

To use your customer managed key with your Amazon Q resources, the following API operations must be permitted in the key policy:

- **kms:CreateGrant** – Adds a grant to a customer managed key. Grants control access to a specified KMS key, which allows access to grant operation Amazon Q requires. For more information about Using Grants, see the AWS Key Management Service Developer Guide.

  This allows Amazon Q to do the following:

  - Call GenerateDataKeyWithoutPlainText to generate an encrypted data key and store it, because the data key isn't immediately used to encrypt.
  - Call Decrypt to use the stored encrypted data key to access encrypted data.
  - Set up a retiring principal to allow the service to RetireGrant.
  - **kms:DescribeKey** – Provides the customer managed key details to allow Amazon Q to validate the key.

  The following are policy statement examples you can add for Amazon Q
"Statement": [{
   "Sid": "Allow access to principals authorized to use Amazon Q",
   "Effect": "Allow",
   "Principal": {
       "AWS": "*"
   },
   "Action": [
       "kms:DescribeKey",
       "kms:CreateGrant"
   ],
   "Resource": "*",
   "Condition": {
       "StringEquals": {
           "kms:ViaService": "qbusiness.region.amazonaws.com",
           "kms:CallerAccount": "111122223333"
       }
   }
},
{"Sid": "Allow access for key administrators",
 "Effect": "Allow",
 "Principal": {
     "AWS": "arn:aws:iam::111122223333:root"
 },
 "Action": [
     "kms:*"
 ],
},
{"Sid": "Allow read-only access to key metadata to the account",
 "Effect": "Allow",
 "Principal": {
     "AWS": "arn:aws:iam::111122223333:root"
 },
 "Action": [
     "kms:Describe*",
     "kms:Get*",
     "kms:List*",
     "kms:RevokeGrant"
 ],
 "Resource": "*"
}
For more information about specifying permissions in a policy, see the AWS Key Management Service Developer Guide.

For more information about troubleshooting key access, see the AWS Key Management Service Developer Guide

**Specifying customer managed key for Amazon Q**

You can specify a customer managed key as a second layer encryption for your Amazon Q application resource.

When you create your application, you can specify the data key by entering a KMS ID, which Amazon Q uses to encrypt the identifiable personal data stored by the application.

**KMS ID** – A key identifier for an AWS KMS customer managed key. Enter a key ID, key ARN, alias name, or alias ARN.

Any resources you create under your Amazon Q application will be encrypted with the same key.

**Monitoring your encryption keys for Amazon Q**

When you use an AWS KMS customer managed key with your Amazon Q resources, you can use AWS CloudTrail or Amazon CloudWatch Logs to track requests that Amazon Q sends to AWS KMS.

The following examples are AWS CloudTrail events for CreateGrant, GenerateDataKey, Decrypt, and DescribeKey to monitor KMS operations called by Amazon Q to access data encrypted by your customer managed key.

**CreateGrant**

When you use an AWS KMS customer managed key to encrypt your application, Amazon Q sends a CreateGrant request on your behalf to access the KMS key in your AWS account. The grant that Amazon Q creates are specific to the resource associated with the AWS KMS customer managed key. In addition, Amazon Q uses the RetireGrant operation to remove a grant when you delete a resource.

The following example event records the CreateGrant operation:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
```

"type": "AssumedRole",
"principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
"arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
"accountId": "111122223333",
"accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
"sessionContext": {
    "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
    },
    "webIdFederationData": {},
    "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-04-22T17:02:00Z"
    }
},
"invokedBy": "qbusiness.amazonaws.com"
},
"eventTime": "2021-04-22T17:07:02Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-west-2",
"sourceIPAddress": "172.12.34.56",
"userAgent": "ExampleDesktop/1.0 (V1; OS)",
"requestParameters": {
    "retiringPrincipal": "qbusiness.region.amazonaws.com",
    "operations": [
        "CreateGrant",
        "RetireGrant",
        "GenerateDataKey",
        "GenerateDataKeyWithoutPlaintext",
        "Encrypt",
        "ReEncryptTo",
        "ReEncryptFrom",
        "Decrypt",
        "DescribeKey"
    ],
    "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "granteePrincipal": "qbusiness.region.amazonaws.com"
GenerateDataKey

When you use an AWS KMS customer managed key for your application, Amazon Q creates a unique table key. It sends a GenerateDataKey request to AWS KMS that specifies the AWS KMS customer managed key for the application.

The following example event records the GenerateDataKey operation:

```json
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "qbusiness.amazonaws.com"
    },
    "eventTime": "2023-11-24T01:50:25Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "Data encryption": 1062
    }
}
```
Decrypt

When you access an encrypted application, Amazon Q calls the Decrypt operation to use the stored encrypted data key to access the encrypted data.

The following example event records the Decrypt operation.

```json
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "qbusiness.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:10:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {  
```

DescribeKey

Amazon Q uses the DescribeKey operation to verify if the AWS KMS customer managed key associated with your application exists in the account and region.

The following example event records DescribeKey operation:

```json
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01"
            }
        }
    },
    "eventSource": "aws.kms",
    "eventSourceARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "eventTime": "2023-04-01T12:00:00Z",
    "requestid": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "awsApiCallDetails": {
        "eventSource": "aws.kms",
        "eventSourceARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "eventName": "DescribeKey",
        "requestParameters": {
            "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
            "EncryptionAlgorithm": "SYMMETRIC_DEFAULT"
        }
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "sharedEventID": "dc129381-1d94-49bd-b522-f56a3482d088"
}
```
Encryption in transit

Amazon Q 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.
Key management

Amazon Q 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.
- A customer-managed KMS key. You can create the key when you are creating an Amazon Q application, retriever, index, web experience, data source, or plugins, or you can create the key using the AWS KMS console. Select a symmetric encryption customer-managed KMS key. Amazon Q does not support asymmetric KMS keys. For more information, see Using Symmetric and Asymmetric Keys in the AWS Key Management Service Developer Guide.

Identity and access management for Amazon Q

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 Q resources. IAM is an AWS service that you can use with no additional charge.

Topics

- **Audience**
- **Authenticating with identities**
- **Managing access using policies**
- **How Amazon Q works with IAM**
- **Identity-based policy examples for Amazon Q**
- **Troubleshooting Amazon Q identity and access**

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Amazon Q.

**Service user** – If you use the Amazon Q service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Amazon Q 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 Q, see Troubleshooting Amazon Q identity and access.

**Service administrator** – If you're in charge of Amazon Q resources at your company, you probably have full access to Amazon Q. It's your job to determine which Amazon Q 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 Q, see How Amazon Q 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 Q. To view example Amazon Q identity-based policies that you can use in IAM, see Identity-based policy examples for Amazon Q.

**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 Signing AWS 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 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.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A federated identity is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the AWS IAM Identity Center 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 When to create an IAM user (instead of a role) 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. You can temporarily assume an IAM role in the AWS Management Console by switching roles. 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 Using IAM roles 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 Creating a role for a third-party Identity Provider 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 How IAM roles differ from resource-based policies 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 [Creating 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 [Using an IAM role to grant permissions to applications running on Amazon EC2 instances](#) in the IAM User Guide.

To learn whether to use IAM roles or IAM users, see [When to create an IAM role (instead of a user)](#) 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 [Creating IAM 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 [Choosing 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 How SCPs work 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 Q works with IAM**

Before you use IAM to manage access to Amazon Q, learn what IAM features are available to use with Amazon Q.

### IAM features you can use with Amazon Q

<table>
<thead>
<tr>
<th>IAM feature</th>
<th>Amazon Q support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity-based policies</td>
<td>Yes</td>
</tr>
<tr>
<td>Resource-based policies</td>
<td>No</td>
</tr>
<tr>
<td>Policy actions</td>
<td>Yes</td>
</tr>
<tr>
<td>Policy resources</td>
<td>Yes</td>
</tr>
<tr>
<td>Policy condition keys</td>
<td>Yes</td>
</tr>
<tr>
<td>ACLs</td>
<td>No</td>
</tr>
<tr>
<td>ABAC (tags in policies)</td>
<td>Yes</td>
</tr>
<tr>
<td>Temporary credentials</td>
<td>Yes</td>
</tr>
<tr>
<td>Principal permissions</td>
<td>Yes</td>
</tr>
<tr>
<td>Service roles</td>
<td>Yes</td>
</tr>
<tr>
<td>Service-linked roles</td>
<td>No</td>
</tr>
</tbody>
</table>

To get a high-level view of how Amazon Q and other AWS services work with most IAM features, see [AWS services that work with IAM](#) in the IAM User Guide.
Identity-based policies for Amazon Q

Supports identity-based policies | Yes

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 [Creating IAM policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorials_create-iam-policy.html) in the [IAM User Guide](https://docs.aws.amazon.com/IAM/latest/UserGuide/)

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. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see [IAM JSON policy elements reference](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements.html) in the [IAM User Guide](https://docs.aws.amazon.com/IAM/latest/UserGuide/).

Identity-based policy examples for Amazon Q

To view examples of Amazon Q identity-based policies, see [Identity-based policy examples for Amazon Q](#).

Resource-based policies within Amazon Q

Supports resource-based policies | No

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.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant
the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see [How IAM roles differ from resource-based policies](#) in the IAM User Guide.

**Policy actions for Amazon Q**

<table>
<thead>
<tr>
<th>Supports policy actions</th>
<th>Yes</th>
</tr>
</thead>
</table>

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.

To see a list of Amazon Q actions, see [Actions Defined by Amazon Q](#) in the Service Authorization Reference.

Policy actions in Amazon Q use the following prefix before the action:

`qbusiness`

Policy actions in Amazon Q use the following prefix before the action: `qbusiness:`. For example, to grant someone permission to list an Amazon Q application with the `ListApplications` API operation, you include the `qbusiness:ListIndices` action in their policy. Policy statements must include either an **Action** or **NotAction** element. Amazon Q 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.

```json
"Action": [
    "qbusiness:action1",
    "qbusiness:action2"
]
```
To view examples of Amazon Q identity-based policies, see [Identity-based policy examples for Amazon Q](#).

**Policy resources for Amazon Q**

| Supports policy resources | Yes |

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)](https://aws.amazon.com). 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": "*
```

To view examples of Amazon Q identity-based policies, see [Identity-based policy examples for Amazon Q](#).

**Policy condition keys for Amazon Q**

| Supports service-specific policy condition keys | Yes |

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.

To see a list of Amazon Q condition keys, see Condition Keys for Amazon Q in the Service Authorization Reference. To learn with which actions and resources you can use a condition key, see Actions Defined by Amazon Q.

To view examples of Amazon Q identity-based policies, see Identity-based policy examples for Amazon Q.

**ACLs in Amazon Q**

<table>
<thead>
<tr>
<th>Supports ACLs</th>
<th>No</th>
</tr>
</thead>
</table>

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.

**ABAC with Amazon Q**

<table>
<thead>
<tr>
<th>Supports ABAC (tags in policies)</th>
<th>Yes</th>
</tr>
</thead>
</table>

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called tags. You can attach tags to IAM entities (users or...
roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is Yes for the service. If a service supports all three condition keys for only some resource types, then the value is Partial.

For more information about ABAC, see [What is ABAC?](#) in the IAM User Guide. To view a tutorial with steps for setting up ABAC, see [Use attribute-based access control (ABAC)](#) in the IAM User Guide.

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.

<table>
<thead>
<tr>
<th>Action</th>
<th>Resource type</th>
<th>Condition keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateApplication</td>
<td></td>
<td><code>aws:ResourceTag</code>,</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>aws:RequestTag</code>,</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>aws:TagKeys</code></td>
</tr>
<tr>
<td>CreateDataSource</td>
<td></td>
<td><code>aws:ResourceTag</code>,</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>aws:RequestTag</code>,</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>aws:TagKeys</code></td>
</tr>
<tr>
<td>CreateIndex</td>
<td></td>
<td><code>aws:ResourceTag</code>,</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>aws:RequestTag</code>,</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>aws:TagKeys</code></td>
</tr>
<tr>
<td>CreatePlugin</td>
<td></td>
<td><code>aws:ResourceTag</code>,</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>aws:RequestTag</code>,</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>aws:TagKeys</code></td>
</tr>
</tbody>
</table>
### Action

<table>
<thead>
<tr>
<th>Action</th>
<th>Resource type</th>
<th>Condition keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateRetriever</td>
<td></td>
<td>aws:ResourceTag, aws:RequestTag,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws:TagKeys</td>
</tr>
<tr>
<td>CreateWebExperience</td>
<td></td>
<td>aws:ResourceTag, aws:RequestTag,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws:TagKeys</td>
</tr>
<tr>
<td>ListTagsForResource</td>
<td>application, index, retriever, data</td>
<td>aws:ResourceTag, aws:RequestTag,</td>
</tr>
<tr>
<td></td>
<td>source, web experience, plugin</td>
<td>aws:TagKeys</td>
</tr>
<tr>
<td>TagResource</td>
<td>application, index, retriever, data</td>
<td>aws:ResourceTag, aws:RequestTag,</td>
</tr>
<tr>
<td></td>
<td>source, web experience, plugin</td>
<td>aws:TagKeys</td>
</tr>
<tr>
<td>UntagResource</td>
<td>application, index, retriever, data</td>
<td>aws:TagKeys</td>
</tr>
<tr>
<td></td>
<td>source, web experience, plugin</td>
<td></td>
</tr>
</tbody>
</table>

For information about tagging Amazon Q resources, see [Tagging resources](#). 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](#).

### Using temporary credentials with Amazon Q

<table>
<thead>
<tr>
<th>Supports temporary credentials</th>
<th>Yes</th>
</tr>
</thead>
</table>

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see [AWS services that work with IAM](#) in the [IAM User Guide](#).

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your
company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switching to a role (console) in the IAM User Guide.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

**Cross-service principal permissions for Amazon Q**

<table>
<thead>
<tr>
<th>Supports forward access sessions (FAS)</th>
<th>Yes</th>
</tr>
</thead>
</table>

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 roles for Amazon Q**

<table>
<thead>
<tr>
<th>Supports service roles</th>
<th>Yes</th>
</tr>
</thead>
</table>

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 Creating a role to delegate permissions to an AWS service in the IAM User Guide.

⚠️ **Warning**

Changing the permissions for a service role might break Amazon Q functionality. Edit service roles only when Amazon Q provides guidance to do so.
Service-linked roles for Amazon Q

| Supports service-linked roles | No |

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.

For details about creating or managing service-linked roles, see AWS services that work with IAM. Find a service in the table that includes a Yes in the Service-linked role column. Choose the Yes link to view the service-linked role documentation for that service.

Identity-based policy examples for Amazon Q

By default, users and roles don’t have permission to create or modify Amazon Q resources. They also can’t perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. 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.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Creating IAM policies in the IAM User Guide.

For details about actions and resource types defined by Amazon Q, including the format of the ARNs for each of the resource types, see Actions, Resources, and Condition Keys for Amazon Q in the Service Authorization Reference.

Topics

- Policy best practices
- Using the Amazon Q console
- Allow users to view their own permissions
- Allow a user to converse with Amazon Q
- Allow an admin to manage plugins in an application
- Allow an admin to manage a specific plugin
- Tag-based policy examples
Policy best practices

Identity-based policies determine whether someone can create, access, or delete Amazon Q 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 *IAM Access Analyzer policy validation* 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 *Configuring MFA-protected API access* in the *IAM User Guide*.

For more information about best practices in IAM, see *Security best practices in IAM* in the *IAM User Guide*.
Using the Amazon Q console

To access the Amazon Q console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Amazon Q resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the Amazon Q console, also attach the Amazon Q ConsoleAccess or ReadOnly AWS managed policy to the entities. For more information, see Adding permissions to a user in the IAM User Guide.

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.

```json
{
   "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",
      
```
Allow a user to converse with Amazon Q

This example allows a user to start conversations with Amazon Q, view past conversations, and delete their conversation history for a specific Amazon Q application. The IAM context key `qbusiness:userId` is used to restrict permissions to a specific user.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "qbusiness:ChatSync",
        "qbusiness:ListMessages",
        "qbusiness:ListConversations",
        "qbusiness:DescribeExperience",
        "qbusiness:DeleteConversation"
      ],
      "Resource": [
        "arn:aws:qbusiness:<REGION>::<ACCOUNT>:application/<APPLICATION_ID>"
      ],
      "Condition": {
        "StringEquals": {
          "qbusiness:userId": "<USER_ID>"
        }
      }
    }
  ]
}
```
Allow an admin to manage plugins in an application

This example allows an Amazon Q admin to manage plugins in a chat application.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "qbusiness:CreatePlugin",
                "qbusiness:ListPlugins",
                "qbusiness:GetPlugin",
                "qbusiness:UpdatePlugin",
                "qbusiness:DeletePlugin"
            ],
            "Resource": [
                "arn:aws:qbusiness:<REGION>::<ACCOUNT>:application/<APPLICATION_ID>"
            ]
        }
    ]
}
```

Allow an admin to manage a specific plugin

This example allows an Amazon Q admin to manage a specific plugin.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "qbusiness:GetPlugin",
                "qbusiness:UpdatePlugin",
                "qbusiness:DeletePlugin"
            ],
            "Resource": [
                "arn:aws:qbusiness:<REGION>::<ACCOUNT>:application/<APPLICATION_ID>",
                "arn:aws:qbusiness:<REGION>::<ACCOUNT>:application/<APPLICATION_ID>/plugin/<PLUGIN_ID>"
            ]
        }
    ]
}
```
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 ChatSync operation with any resource tagged with the key `department` and the value `finance`.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "qbusiness:ChatSync"
            ],
            "Resource": [ "*" ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/department": "finance"
                }
            }
        }
    ]
}
```

Example: Use a tag to activate operations

This example policy grants a user or role in your AWS account permission to use any Amazon Q operation except the TagResource operation with any resource tagged with the key `department` and the value `finance`.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "qbusiness:*",
            "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 ChatSync 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": "qbusiness:ChatSync",
        "Resource": "*",
    },
    {
        "Effect": "Deny",
        "Action": "qbusiness:ChatSync",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/department": "finance"
            }
        }
    },
    {
        "Effect": "Deny",
        "Action": "qbusiness:ChatSync",
        "Resource": ""
    }
}
```
Troubleshooting Amazon Q identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Amazon Q and IAM.

Topics

- I am not authorized to perform an action in Amazon Q
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my Amazon Q resources

I am not authorized to perform an action in Amazon Q

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the mateojackson IAM user tries to use the console to view details about a fictional `my-example-widget` resource but doesn't have the fictional qbusiness:`GetWidget` permissions.

User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: qbusiness:GetWidget on resource: `my-example-widget`

In this case, the policy for the mateojackson user must be updated to allow access to the `my-example-widget` resource by using the qbusiness:`GetWidget` action.
If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

**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 Q.

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 Q. 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.

```plaintext
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 want to allow people outside of my AWS account to access my Amazon Q 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 Q supports these features, see [How Amazon Q 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 How IAM roles differ from resource-based policies in the IAM User Guide.

Compliance validation for Amazon Q

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. 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 AWS services 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 baseline environments on AWS that are security and compliance focused.

• Architecting for HIPAA Security and Compliance on Amazon Web Services – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

Note
Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.

• AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.

• AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- **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. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see [Security Hub controls reference](#).

- **AWS Audit Manager** – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

## Resilience in Amazon Q

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](#).

## Infrastructure security in Amazon Q

As a managed service, Amazon Q is protected by the AWS global network security procedures that are described in the *Amazon Web Services: Overview of Security Processes* whitepaper.

You use AWS published API calls to access Amazon Q through the network. Clients must support the following:

- Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later.
- 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.
Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the calling service) calls another service (the called service). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in resource policies to limit the permissions that Amazon Q gives another service to the resource. Use `aws:SourceArn` if you want only one resource to be associated with the cross-service access. Use `aws:SourceAccount` if you want to allow any resource in that account to be associated with the cross-service use.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full Amazon Resource Name (ARN) of the resource. If you don't know the full ARN of the resource or if you're specifying multiple resources, use the `aws:SourceArn` global context condition key with wildcard characters (*) for the unknown portions of the ARN. For example, `arn:aws:qbusiness:*:123456789012:*`.

If the `aws:SourceArn` value doesn't contain the account ID, such as an Amazon S3 bucket ARN, you must use both global condition context keys to limit permissions.

The value of `aws:SourceArn` must be `ResourceDescription`.

The following example shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in Amazon Q to prevent the confused deputy problem.

```json
{
   "Version": "2012-10-17",
   "Statement": {
      "Sid": "ConfusedDeputyPreventionExamplePolicy",
      "Effect": "Allow",
      "Principal": {
         "Service": "qbusiness.amazonaws.com"
      }
   }
}
```
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**

Security best practices

Amazon Q provides several 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 Q 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 Q applications.
Monitoring Amazon Q

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon Q and your other AWS solutions. AWS provides the following monitoring tools to watch Amazon Q, report when something is wrong, and take automatic actions when appropriate:

- **AWS CloudTrail** captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the [AWS CloudTrail User Guide](#).

- **Amazon CloudWatch** monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the [Amazon CloudWatch User Guide](#).

- **Amazon CloudWatch Logs** enables you to monitor, store, and access your log files from Amazon EC2 instances, CloudTrail, and other sources. CloudWatch Logs can monitor information in the log files and notify you when certain thresholds are met. You can also archive your log data in highly durable storage. For more information, see the [Amazon CloudWatch Logs User Guide](#).

Topics

- Logging Amazon Q API calls using AWS CloudTrail

Logging Amazon Q API calls using AWS CloudTrail

Amazon Q is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon Q. CloudTrail captures all API calls for Amazon Q as events. The calls captured include calls from the Amazon Q console and code calls to the Amazon Q API operations. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon Q. 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 Q, the IP address from which the request was made, who made the request, when it was made, and additional details.
For more information about CloudTrail, including how to configure and activate it, see the AWS CloudTrail User Guide.

Amazon Q information in CloudTrail

CloudTrail is activated on your AWS account when you create the account. When activity occurs in Amazon Q, that activity is recorded in a CloudTrail event along with other AWS service events in Event history. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail Event history in the AWS CloudTrail User Guide.

For an ongoing record of events in your AWS account, including events for Amazon Q, create a trail. A trail enables CloudTrail to deliver log files to an Amazon 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 Amazon 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 the following topics:

- Creating a trail for your AWS account
- 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

Control plane events in CloudTrail

CloudTrail supports logging the following Amazon Q actions documented in the Amazon Q API Reference:

- CreateApplication
- DeleteApplication
- GetApplication
- ListApplications
- UpdateApplication
- DeleteChatControlsConfiguration
- GetChatControlsConfiguration
- UpdateChatControlsConfiguration
• CreateDataSource
• DeleteDataSource
• GetDataSource
• ListDataSources
• UpdateDataSource
• CreateWebExperience
• DeleteWebExperience
• ListWebExperiences
• UpdateWebExperience
• CreateIndex
• DeleteIndex
• GetIndex
• ListIndices
• UpdateIndex
• CreatePlugin
• DeletePlugin
• GetPlugin
• ListPlugins
• UpdatePlugin
• CreateRetriever
• DeleteRetriever
• GetRetriever
• ListRetrievers
• UpdateRetriever
• ListTagsForResource
• TagResource
• UntagResource

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:
• Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
• Whether the request was made with temporary security credentials for a role or federated user.
• Whether the request was made by another AWS service.

For more information, see CloudTrail userIdentity element in the AWS CloudTrail User Guide.

Data plane events in CloudTrail

Data events provide information about the resource operations performed on or in a resource (for example, reading or writing to an Amazon S3 object). These are also known as data plane operations. By default, CloudTrail doesn't log data events.

The following table shows the Amazon Q API operations logged to CloudTrail as data events. The Data event type (console) column shows the appropriate selection in the CloudTrail console. The Amazon Q resource types column shows the resources.type value that you would specify to log data events for the resource.

<table>
<thead>
<tr>
<th>Data event type (console)</th>
<th>Amazon Q resource types</th>
<th>Supported data events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Q Business application</td>
<td>AWS::QBusiness::Application</td>
<td>• ListDataSourceSyncJobs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• StartDataSourceSyncJob</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• StopDataSourceSyncJob</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• BatchPutDocument</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• BatchDeleteDocument</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PutFeedback</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ChatSync</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DeleteConversation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ListConversations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ListMessages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ListGroups</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DeleteGroup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• GetGroup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PutGroup</td>
</tr>
<tr>
<td>Data event type (console)</td>
<td>Amazon Q resource types</td>
<td>Supported data events</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------</td>
<td>-----------------------</td>
</tr>
</tbody>
</table>
|                          | AWS::QBusiness::DataSource | • CreateUser  
|                          |                        | • DeleteUser  
|                          |                        | • GetUser  
|                          |                        | • UpdateUser  
|                          |                        | • ListDocuments  
| Amazon Q Business data resource | AWS::QBusiness::DataSource | • ListDataSourceSyncJobs  
|                          |                        | • StartDataSourceSyncJob  
|                          |                        | • StopDataSourceSyncJob  
| Amazon Q Business index | AWS::QBusiness::Index | • DeleteGroup  
|                          |                        | • GetGroup  
|                          |                        | • PutGroup  
|                          |                        | • ListGroups  
|                          |                        | • ListDocuments  
|                          |                        | • BatchPutDocument  
|                          |                        | • BatchDeleteDocument  

You can log these API operations by configuring advanced event selectors to record data events for the Amazon Q resource types: AWS::QBusiness::Application, AWS::QBusiness::DataSource, and AWS::QBusiness::Index. To configure advanced event selectors, you can use either the CloudTrail console or the AWS CLI:

- From the CloudTrail console, choose the **Data event type** for which you want to log data events. Additionally, you can filter on the `eventName` and `resources.ARN` fields by choosing a custom log selector template. For more information, see [Logging data events with the AWS Management Console](https://docs.aws.amazon.com/AmazonCloudTrail/latest/userguide/logging-data-events-with-the-amazon-cloudtrail-management-console.html) in the *AWS CloudTrail User Guide*.

- From the AWS CLI, specify the `resources.type` value for which you want to log data events and set the `eventCategory` equal to `Data`. For more information, see [Logging data events with the AWS CLI](https://docs.aws.amazon.com/AmazonCloudTrail/latest/userguide/logging-data-events-with-the-amazon-cloudtrail-command-line-interface.html) in the *AWS CloudTrail User Guide*.

The following example shows how to configure a trail to log all Amazon Q data events for all Amazon Q resource types.
aws cloudtrail put-event-selectors --trail-name trailName \
--advanced-event-selectors \ 
'[ \\
   "Name": "Log all data events on an Amazon Q application", 
   "FieldSelectors": [ 
      { "Field": "eventCategory", "Equals": ["Data"] }, 
      { "Field": "resources.type", "Equals": ["AWS::QBusiness::Application"] } 
   ], 
   "Name": "Log all data events on an Amazon Q data source", 
   "FieldSelectors": [ 
      { "Field": "eventCategory", "Equals": ["Data"] }, 
      { "Field": "resources.type", "Equals": ["AWS::QBusiness::DataSource"] } 
   ], 
   "Name": "Log all data events on an Amazon Q index", 
   "FieldSelectors": [ 
      { "Field": "eventCategory", "Equals": ["Data"] }, 
      { "Field": "resources.type", "Equals": ["AWS::QBusiness::Index"] } 
   ] 
]
'

You can additionally filter on the eventName and resources.ARN fields. For more information about configuring these fields, see AdvancedFieldSelector in the AWS CloudTrail API Reference.

Additional charges apply for data events. For more information about CloudTrail pricing, see AWS CloudTrail Pricing.

Amazon Q management events in CloudTrail

Management events provide information about management operations that are performed on resources in your AWS account. These management events are also known as control plane operations. CloudTrail logs management event API operations by default.

Amazon Q logs the remainder of Amazon Q API operations as management events. For a list of the Amazon Q API operations that Amazon Q logs to CloudTrail, see the Amazon Q API Reference.
Understanding Amazon Q log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. 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.

The following example shows a CloudTrail log entry that demonstrates the `CreateApplication` action.

```json
{
   "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": "qbusiness.amazonaws.com",
   "eventName": "CreateApplication",
   "awsRegion": "region",
   "sourceIPAddress": "region",
   "userAgent": "user agent",
   "requestParameters": {
      "name": "name",
   }
}
```
"roleArn": "description",
   "clientToken": "client token"
 },
"responseElements": {
   "applicationId": "application ID"
 },
"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": "qbusiness.us-west-2.api.aws"
  }
}
Service quotas for Amazon Q

The following are the service endpoints and service quotas for Amazon Q. To connect programmatically to Amazon Q, you use an endpoint. For more information, see AWS service endpoints in the AWS General Reference. Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account. For more information, see AWS service quotas in the AWS General Reference.

Supported Regions

The following table shows the AWS Regions and endpoints currently supported by Amazon Q.

<table>
<thead>
<tr>
<th>Region name</th>
<th>Region</th>
<th>Endpoint</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (N. Virginia)</td>
<td>us-east-1</td>
<td>qbusiness.us-east-1.api.aws</td>
<td>HTTPS</td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td>us-west-2</td>
<td>qbusiness.us-west-2.api.aws</td>
<td>HTTPS</td>
</tr>
</tbody>
</table>

Quotas

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, and other quotas can't be increased.

To view the quotas for Amazon Q, open the Service Quotas console. In the navigation pane, choose AWS services and select Amazon Q.

Some service quotas can be adjusted or increased. To see whether a quota can be adjusted, refer to the Adjustable column in the following table. To request a quota increase, use the limit increase form.

The following table shows the quotas that are related to Amazon Q for your AWS account.
<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Adjustable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications</td>
<td>50</td>
<td>No</td>
<td>Maximum number of applications per account.</td>
</tr>
<tr>
<td>Data sources</td>
<td>50</td>
<td>No</td>
<td>Maximum number of data sources per application.</td>
</tr>
<tr>
<td>Plugins</td>
<td>4</td>
<td>No</td>
<td>Maximum number of plugins per application.</td>
</tr>
</tbody>
</table>
API reference

For more information on Amazon Q APIs, see the Amazon Q API reference.
AWS Glossary

For the latest AWS terminology, see the AWS glossary in the AWS Glossary Reference.
Document history

The following table describes the documentation releases for Amazon Q.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preview release</td>
<td>This is the initial preview release of the Amazon Q (For Business Use) Developer Guide.</td>
<td>November 28, 2023</td>
</tr>
</tbody>
</table>