

Reimagine search on GitHub repositories with the power of the Amazon Kendra GitHub connector

by Manjula Nagineni and Arjun Agrawal | on 02 JUN 2022 | in [Amazon Kendra](#), [Artificial Intelligence](#) | [Permalink](#) | [Comments](#) | [Share](#)

[Amazon Kendra](#) offers highly accurate semantic and natural language search powered by machine learning (ML).

Many organizations use GitHub as a code hosting platform for version control and to redefine collaboration of open-source software projects. A GitHub account repository might include many content types, such as files, issues, issue comments, issue comment attachments, pull requests, pull request comments, pull request comment attachments, and more. This corpus data is scattered across multiple locations and content repositories (public, private, and internal) within an organization. However, surfacing the relevant information in a traditional keyword search is ineffective. You can now use the new Amazon Kendra data source for GitHub to index specific content types and easily find information from this data. The GitHub data source syncs the data in your GitHub repositories to your Amazon Kendra index.

This post guides you through the step-by-step process to configure the Amazon Kendra connector for GitHub. We also show you how to configure for the connector both GitHub Enterprise Cloud (SaaS) and GitHub Enterprise Server (on premises) services.

Solution overview

The solution consists of the following high-level steps:

1. Set up your GitHub enterprise account.
2. Set up a GitHub repo.
3. Create a GitHub data source connector.
4. Search the indexed content.

Prerequisites

You need the following prerequisites to set up the Amazon Kendra connector for GitHub:

- An [AWS account](#) with privileges to create [AWS Identity and Access Management](#) (IAM) roles and policies. For more information, see [Overview of access management: Permissions and policies](#).
- Basic knowledge of AWS and working knowledge of GitHub enterprise products. For more details, see [Getting started with GitHub Enterprise Cloud](#) and [Getting started with GitHub Enterprise Server](#).
- Enterprise owner or administrator access to a GitHub enterprise account.
- [Personal access tokens](#) for authentication to GitHub. For more information, refer to [Creating an OAuth App](#), [Authorizing OAuth Apps](#), and [Scopes for OAuth Apps](#).

- An [AWS Secrets Manager](#) secret to store your GitHub authentication credentials. For more information, see [Using a GitHub data source](#).

Set up your GitHub enterprise account

Create an [enterprise account](#) before proceeding to the next steps. For authentication, you can specify two types of tokens while configuring the GitHub connector:

- **Personal access token** – Direct API requests that you authenticate with a personal access token are user-to-server requests. User-to-server requests are limited to 5,000 requests per hour and per authenticated user. Your personal access token is also an OAuth token.
- **OAuth token** – With this token, the requests are subject to a higher limit of 15,000 requests per hour and per authenticated user.

Our recommendation is to use an OAuth token for better API throttle limits and connector performance.

For this post, we assume you have an enterprise account and generated OAuth token.

Set up your GitHub repo

To configure your GitHub repo, complete the following steps:

1. Create a new repository, and specify its owner and name.


2. Choose if the repository is public, internal, or private.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

Repository name *

 [redacted]-Connector ▾


 /

kendra-githubconnector-demo ✓


Great repository names are short and memorable. Need inspiration? How about [supreme-octo-guide?](#)

Description (optional)


Demo purpose

☐  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☒  **Internal**

Persistent Systems - EA 2 [enterprise members](#) can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

3. For this post, update the README file with the following text:

```
CreateIndex API creates a new Amazon Kendra index. Index creation is an asynchronous  
Once the index is active you can index your documents using the BatchPutDocument API
```

4. You can add a sample file to your repository with commit changes. The following is an example of using Amazon Kendra in Python:

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

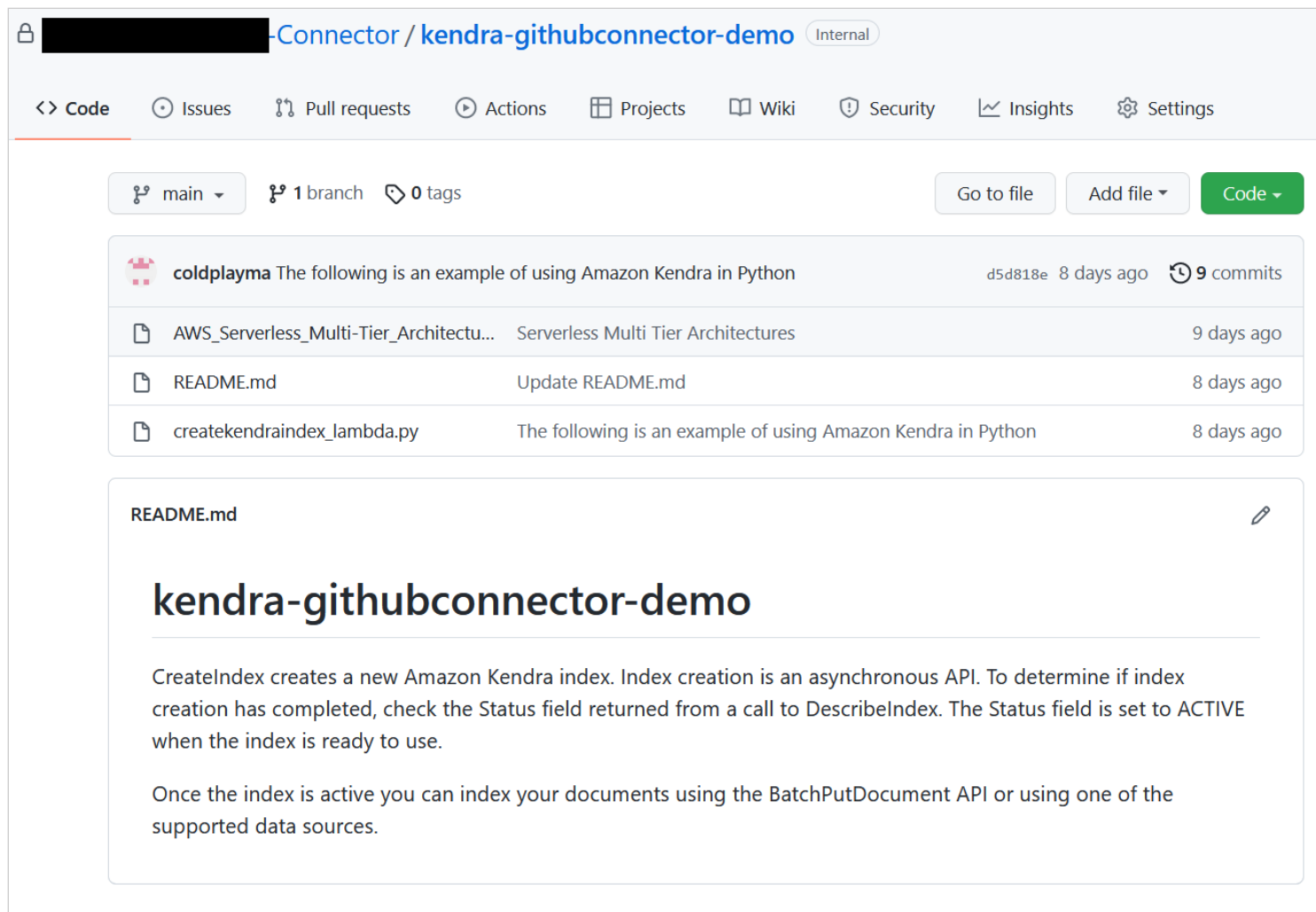
# For this example, there should be one job
status = jobs["History"][0]["Status"]

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

except ClientError as e:
```

5. Download [AWS_Whitepapers.zip](#) to your computer, and extract the files into a folder called `AWS_Whitepaper`.
6. Upload `AWS Whitepapers/Best Practices/AWS Serverless Multi-Tier Architectures` to your repository.

Your repository should look like the following screenshot.



The screenshot shows a GitHub repository page for 'kendra-githubconnector-demo'. The repository is owned by 'coldplayma' and contains a README.md file and a Python script 'createkendraindex_lambda.py'. The README.md file contains the following text:

kendra-githubconnector-demo

CreateIndex creates a new Amazon Kendra index. Index creation is an asynchronous API. To determine if index creation has completed, check the Status field returned from a call to DescribeIndex. The Status field is set to ACTIVE when the index is ready to use.

Once the index is active you can index your documents using the BatchPutDocument API or using one of the supported data sources.

Your organization's code repositories might hold hundreds of thousands of documents, README notes, code comments, webpages, and other items. In the next section, we showcase the document comprehension capability of Amazon Kendra to find the relevant information contained in these repositories.

Create a GitHub data source connector

For this post, we assume you have already created an Amazon Kendra index. If you don't have an index, [create a new index](#) before proceeding with the following steps.

1. On the Amazon Kendra console, choose the index that you want to add the data source to.
2. Choose **Add data sources**.

3. From the list of data source connectors, choose **Add connector** under **GitHub**.

Select sample dataset (Amazon S3 data source)

Sample AWS documentation
Covers Kendra, EC2, S3, and Lambda



Add dataset

Select data source connector type [Info](#)

Amazon FSx

FSx

5 steps to complete

Add connector

Amazon RDS



4 steps to complete

Add connector

Amazon S3



3 steps to complete

Add connector

Confluence

 Confluence

5 steps to complete

Add connector

Custom data source connector



3 steps to complete

Add connector

GitHub



5 steps to complete

Add connector

4. On the **Specify data source details** page, enter a data source name and an optional description.

5. To assign metadata to your AWS resources in the form of tags, choose **Add tags** and enter a key and value.

6. Choose Next.

Amazon Kendra > Indexes > github-index > Data sources > Add data source > GitHub

Step 1
Specify data source details

Step 2
Define access and security

Step 3
Configure sync settings

Step 4
Set field mappings - optional

Step 5
Review and create

Specify data source details [Info](#)

Name and description

Data source name

github-datasource

Maximum of 1000 alphanumeric characters. Can include hyphens (-), but not spaces.

Description - optional

This text is viewed only by Amazon Kendra administrators and can be edited later.

Data source for GitHub enterprise

Language [Info](#)

Default language

Select a language to ingest documents (defaults to English). Language specified in document metadata overrides selected language.

English (en)

Tags (1) - optional [Info](#)

A tag is an administrative label that you assign to AWS resources to make it easier to manage them. Each tag consists of a key and an optional value. Use tags to search and filter your resources or track your AWS costs.

Key

Name

Value - optional

github-connector

Remove

Add new tag

You can add up to 49 more tags.

Cancel

Next

7. On the **Define access and security** page, choose your GitHub source. Amazon Kendra supports two types of GitHub services:

- GitHub Enterprise Cloud** – If you choose this option, specify the GitHub host URL and GitHub organization name. Configure your Secrets Manager secret with the authentication credentials in the form of an OAuth2 access token of the GitHub enterprise owner. The OAuth2 token scope should be authorized for

`repo:status` , `public_repo` , `repo:invite` , `read:org` , `user:email` , and `read:user` .

Define access and security [Info](#)

Source

GitHub source [Info](#)
This is your type of GitHub service such as GitHub Enterprise Cloud or GitHub Enterprise Server.

GitHub Enterprise Cloud ▼

GitHub host URL
Enter the GitHub host name with the protocol(`http://` or `https://`)

`https://api.github.com`

GitHub organization name
Log in to GitHub desktop and go to Your organizations under your profile picture dropdown.

██████████-Connector

Authentication [Info](#)

Authentication credentials
You use an AWS Secrets Manager secret to store your GitHub authentication credentials. If you have a secret containing your credentials, you can use it. Otherwise, create one.

AWS Secrets Manager secret
Choose an existing secret that starts with 'AmazonKendra-' or create a new one.

AmazonKendra-GitHub-cloud01 ▼

- b. **GitHub Enterprise Server** – If you choose this option, specify the GitHub host URL and GitHub organization name you created in the previous section. Configure your Secrets Manager secret with the authentication credentials in the form of an OAuth2 access token of the GitHub enterprise owner. The OAuth2 token scope should be authorized for `repo:status` , `public_repo` , `repo:invite` , `read:org` , `user:email` , `read:user` , and `site_admin` . To configure the SSL certificate, you can create a self-signed certificate for this post using `openssl x509 -in sample.pem -out new_github.cer` and add this certificate to an

S3 bucket.

Define access and security Info

Source

GitHub source Info
This is your type of GitHub service such as GitHub Enterprise Cloud or GitHub Enterprise Server.

GitHub Enterprise Server ▼

GitHub host URL
Enter the GitHub host name with the protocol(http:// or https://)

https://[REDACTED].us-west-2.compute.amazonaws.com/api/v3

GitHub organization name
Log in to GitHub desktop and go to Your organizations under your profile picture dropdown.

PersistentKendraProject

SSL Certificate location

s3://[REDACTED]/new_github.cer Browse S3

[Add SSL certificate to S3](#)

Authentication Info

Authentication credentials
You use an AWS Secrets Manager secret to store your GitHub authentication credentials. If you have a secret containing your credentials, you can use it. Otherwise, create one.

AWS Secrets Manager secret
Choose an existing secret that starts with 'AmazonKendra-' or create a new one.

AmazonKendra-GitHub-server01 ▼

8. For **Virtual Private Cloud (VPC)**, choose the default option (**No VPC**).

9. For **IAM role**, choose **Create a new role (recommended)** and enter a role name.

Whenever you modify the Secrets Manager secret, make sure you also modify the IAM role, because it requires permission to access your secret to authenticate your GitHub account. For more information on the required permissions to include in the IAM role, see [IAM roles for data sources](#).

10. Choose Next.

Configure VPC and security group - optional [Info](#)

Virtual Private Cloud (VPC)
Select a VPC that defines the virtual networking environment for this repository instance. [Manage VPCs](#)

No VPC ▼

IAM role [Info](#)

IAM role
Create a new role (Recommended) ▼

Role name
Your role name will be prefixed with 'AmazonKendra-'. The created role will only work for this data source and its specific configuration.

AmazonKendra-github-role

ⓘ IAM roles used for indexes or FAQs can't be used for data sources. If you are unsure if an existing role is used for a data source or FAQ, choose "Create a new role" to avoid an error.

[Cancel](#)
[Previous](#)
[Next](#)

On the **Configure sync settings** page, you provide details about the sync scope and run schedule.

11. For **Select repositories to crawl**, select **Select repositories** to configure a specific list.
12. Choose the repository `kendra-githubconnector-demo` that you created earlier.
13. Optionally, you can adjust the crawl mode. The GitHub connector supports the two modes:
 - a. **Full crawl mode** – It crawls the entire GitHub organization as configured whenever there is a data source sync. By default, the connector runs in this mode.
 - b. **Change log mode** – It crawls the specified changed GitHub content (added, deleted, modified, permission changes) of the organization whenever there is a data source sync.
14. Optionally, you can filter on the specific content types to index, and configure inclusion and exclusion filters on the file name, type, and path.

Configure sync settings [Info](#)

Sync scope [Info](#)

Select repositories to crawl

Select all or specific repositories in your GitHub organization.

☐ All

☒ Select repositories

Select all or specific repositories you want to crawl.

Enter the name of your repositories

▼ Additional configuration (change log) - optional [Info](#)

You can use the change log to update your index instead of scanning all of the files.

☐ Change log mode

▼ Additional configuration (content types) - optional [Info](#)

Include repository files, issue & pull requests, issue & pull request comments, issue & pull request comment attachments.

Content type

▼

► Additional configuration (regex patterns) - optional [Info](#)

Include and exclude documents from indexing.

15. Under **Sync run schedule**, for **Frequency**, choose **Run on demand**.

16. Choose **Next**.

Sync run schedule [Info](#)

Tell Amazon Kendra how often it should sync this data source. You can check the health of your sync jobs in the data source details page once the data source is created.

Frequency

Select how often you want your data source to sync.




▼

17. In the **Set fields mapping** section, define the mappings between GitHub fields to Amazon Kendra field names. You can configure for each content type and enable these GitHub fields as facets to further refine your search results. For this post, we use the default options.

18. Choose **Next**.

Set field mappings - *optional*
Info

▼ Field mapping guide

Default data source fields
Amazon Kendra associates required data source fields with fields in your index. This is called mapping. Amazon Kendra provides default mapping for required fields.

Custom data source fields
To add custom or little-used fields, choose **Add field**, then specify the name of the field, the index field or new index field that it should map to, and the data type. A new field immediately appears in the tables. If you add new index fields, they are added to the index when you create the data source.

Add fields via your API
You can add fields directly with the API.

Repository (8/8)
Add field

<input checked="" type="checkbox"/>	GitHub field name	Description	Index field name	Data type
<input checked="" type="checkbox"/>	Description	Default	_document_body	String
<input checked="" type="checkbox"/>	repositoryName	Custom	<input type="text" value="gh_repository_name"/>	String
<input checked="" type="checkbox"/>	repositoryVisibility	Custom	<input type="text" value="gh_repository_visibility"/>	String
<input checked="" type="checkbox"/>	category	Default	_category	String
<input checked="" type="checkbox"/>	owner	Default	_authors	String list
<input checked="" type="checkbox"/>	sourceUrl	Default	_source_uri	String
<input checked="" type="checkbox"/>	createdAt	Default	_created_at	Date
<input checked="" type="checkbox"/>	updatedAt	Default	_last_updated_at	Date

19. On the **Review and create** page, review your options for the GitHub data source.20. Choose **Add data source**.21. After the data source is created, choose **Sync now** to index the data from GitHub.

Search indexed content

After about 10 minutes, the data source sync is complete and the GitHub content is ingested into the index. The GitHub connector crawls the following entities:

- Repositories on GitHub Enterprise Cloud:
 - Repository with its description
 - Code and their branches with folders and subfolders
 - Issues and pull request files for public repositories
 - Issues and pull request comments and their replies for public and private repositories

- Issues and pull request comment attachments and their replies' attachments for public repositories
- Repositories on GitHub Enterprise Server:
 - Repository with its description
 - Code and their branches with folders and subfolders
 - Issues and pull request comments and their replies for public, private, and internal repositories

Now you can test some queries on the Amazon Kendra Search console.

1. Choose **Search indexed content**.

2. Enter the sample text `How to check the status of the index creation?`

Search console

Q How to check the status of the index creation? X

▶ Test query with user name or groups

1-4 of 4 results

Amazon Kendra suggested answers

[README.md](#)

kendra-githubconnector-demo CreateIndex creates a new Amazon Kendra **index**. **Index creation** is an asynchronous API. **To determine if index creation has completed, check the Status field returned from a call to DescribeIndex. The Status field is set to ACTIVE when the index is ready to use.** Once the **index** is active you can **index** your documents using the BatchPutDocument API or using one of the supported data sources.

[https://github.com/\[redacted\]-Connector/.../main/README.md](https://github.com/[redacted]-Connector/.../main/README.md)

▶ Document fields

👍 🗨️

What are Amazon Kendra suggested answers? [Info](#)

Sort: Relevance ▼ ↓

[README.md](#)

...githubconnector-demo CreateIndex creates a new Amazon Kendra **index**. **Index creation** is an asynchronous API. To determine if **index creation** has completed, **check the Status field** returned from a call to DescribeIndex. The **Status** field is set to ACTIVE when the **index** is ready to use. Once the **index**...

[https://github.com/\[redacted\]-Connector/.../main/README.md](https://github.com/[redacted]-Connector/.../main/README.md)

▶ Document fields

👍 🗨️

[createkendraindex_lambda.py](#)

...index_description = kendra.describe_index(Id = index_id) # When status is not CREATING quit. status = index_description["Status"]
print(" Creating index...

[https://github.com/\[redacted\]-Connector/.../createkendraindex_lambda.py](https://github.com/[redacted]-Connector/.../createkendraindex_lambda.py)

▶ Document fields

👍 🗨️

3. Run another query and enter the sample text **What are most popular usecases for AWS Lambda?**

The screenshot shows the Amazon Kendra Search console interface. At the top, there's a search bar with the query "What are most popular usecases for AWS Lambda?". Below the search bar, there's a button labeled "Test query with user name or groups". The results section shows "1-4 of 4 results". The first result is titled "AWS_Serverless_Multi-Tier_Architectures.pdf" and is labeled as an "Amazon Kendra suggested answer". The snippet describes how AWS Lambda functions can be used for event-driven data processing workflows, such as processing files from Amazon S3 or streaming data from Amazon Kinesis. It mentions that Amazon API Gateway acts as the front door for the logic tier, and AWS Lambda executes the application code. The snippet includes several URLs: <https://aws.amazon.com/lambda/faqs/>, <https://aws.amazon.com/lambda/faqs/>, <https://aws.amazon.com/s3>, <https://aws.amazon.com/kinesis/>, and "Amazon Web Services AWS Serverless Multi-Tier Architectures Page 4". It also mentions "Your Business Logic Goes Here, No Servers Necessary". Below the snippet, there's a link to the PDF document: https://github.com/.../AWS_Serverless_Multi-Tier_Architectures.pdf. There are also "Document fields" and feedback icons (thumbs up and thumbs down). At the bottom of the results section, there's a "Sort: Relevance" dropdown menu and a "What are Amazon Kendra suggested answers? Info" link.

Amazon Kendra accurately surfaces relevant information based on the content indexed from the GitHub repositories. Access control to all the information is still enforced by the original repository.

Clean up

To avoid incurring unnecessary charges, clean up the resources you created for testing this connector.

1. Delete the Amazon Kendra index if you created one specifically for testing this solution.
2. Delete the GitHub connector data source if you added a new data source to an existing index.
3. Delete the content you added for your GitHub account.

Conclusion

In this post, we covered the process of setting up the new Amazon Kendra connector for GitHub. Organizations can empower their software developers by providing secure and intelligent search of content spread across many different GitHub repositories.

This post illustrates the basic connector capabilities. You can also customize the search by enabling facets based on GitHub fields and map to Amazon Kendra index fields. With the GitHub connector, you can control access to the data because it can crawl `orgname-reponame` and set a group as the principle and collaborators of the repository as members of the group. Furthermore, Amazon Kendra provides features such as [Custom Document Enrichment](#) and [Experience Builder](#) to enhance the search experience.

For more details about Amazon Kendra, refer to the [Amazon Kendra Developer Guide](#).

About the Authors



Manjula Nagineni is a Solutions Architect with AWS based in New York. She works with major Financial service institutions, architecting, and modernizing their large-scale applications while adopting AWS cloud services. She is passionate about designing big data workloads cloud-natively. She has over 20 years of IT experience in Software Development, Analytics and Architecture across multiple domains such as finance, manufacturing and telecom.



Arjun Agrawal is Software Development Engineer at AWS Kendra.

Comments

o Comments

1 Logi

G

Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

♡ Share

Best Newest Oldest

Be the first to comment.

Subscribe Privacy Do Not Sell My Data