Learn more

- For an overview of the Amazon Q web experience creation process using IAM Identity Center, see Configuring an application using IAM Identity Center.
- For an overview of the Amazon Q web experience creation process using AWS Identity and Access Management, see Configuring an application using IAM.
- For an overview of connector features, see Data source connector concepts.
- For information about connector configuration best practices, see Connector configuration best practices.

Topics

GitHub (Cloud) connector overview

The following table gives an overview of the Amazon Q Business GitHub (Cloud) connector and its supported features.

Category	Feature	Support
Security	**Authentication type**	Personal token, OAuth token
	Authentication credentials	• GitHub (Cloud) token
	Access Control List (ACL) crawling	Yes. For more information, see ACL crawling.
	Identity crawling	Yes

Category	Feature	Support
	VPC	Yes
Crawl features	Custom metadata	Yes
	Entities	Yes. The following entities are supported: • Repository • Repository Commit • Issue Document • Issue Comment • Issue Attachment • Pull Request Comment • Pull request Document • Pull Request Attachment
	Field mappings	Yes. Supports default and custom field mappings. For more information, see Field mappings.
	Filters	Yes. The following filters are supported: • Include select repositories • Include content by specific entities. • Include specific branched by name • Include/exclude content by file name, file type, and file path
	Sync mode	Supports full and incremental sync.
	File types	Supports all files supported by Amazon Q.

Prerequisites for connecting Amazon Q Business to GitHub (Cloud)

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

In GitHub (Cloud), make sure you have:

- Created a GitHub (Cloud) user with administrative permissions to the GitHub (Cloud) organization.
- Created a classic personal access token for authentication credentials. See GitHub (Cloud) 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 (Cloud) documentation on OAuth authorization.
- Noted the GitHub (Cloud) host URL for the type of GitHub (Cloud) service that you use. For example, the host URL for GitHub (Cloud) Cloud could be *`https://api.github.com`*.
- Noted the name of your organization for GitHub (Cloud) the GitHub (Cloud) Enterprise account you want to connect to. You can find your organization name by logging into GitHub (Cloud) desktop and selecting **Your organizations** under your profile picture dropdown.
- Added the following OAuth scope permissions in GitHub (Cloud):
  - `repo:status` – Grants read/write access to commit statuses in public and private repositories. This scope is only necessary to grant other users or services access to private repository commit statuses without granting access to the code.
  - `repo_deployment` – Grants access to deployment statuses for public and private repositories. This scope is only necessary to grant other users or services access to deployment statuses, without granting access to the code.
  - `public_repo` – Limits access to public repositories. That includes read/write access to code, commit statuses, repository projects, collaborators, and deployment statuses for public repositories and organizations. Also required for starring public repositories.
  - `repo:invite` – Grants accept/decline abilities for invitations to collaborate on a repository. This scope is only necessary to grant other users or services access to invites without granting access to the code.
  - `security_events` – Grants: read and write access to security events in the code scanning API. This scope is only necessary to grant other users or services access to security events without granting access to the code.
  - `read:org` – Read-only access to organization membership, organization projects, and team membership.
  - `user:email` – Grants read access to a user's email addresses. Required by Amazon Q Business to crawl ACLs.

- `user:follow` – Grants access to follow or unfollow other users. Required by Amazon Q Business to crawl ACLs.
- `read:user` – Grants access to read a user's profile data. Required by Amazon Q Business to crawl ACLs.
- `workflow` – Grants the ability to add and update GitHub (Cloud) Actions workflow files. Workflow files can be committed without this scope if the same file (with both the same path and contents) exists on another branch in the same repository.

For more information, see Scopes for OAuth apps in GitHub (Cloud) Docs.

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 (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.

Connecting Amazon Q Business to GitHub (Cloud) using the console

The following procedure outlines how to connect Amazon Q Business to GitHub (Cloud) using the AWS Management Console.

Connecting Amazon Q to GitHub (Cloud)

1. Sign in to the AWS Management Console and open the Amazon Q Business console.
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 50 data sources.
5. Then, on the **GitHub (Cloud)** page, enter the following information:
6. **Name** – Name your data source for easy tracking.

   Note: You can include hyphens (-) but not spaces. Maximum of 1,000 alphanumeric characters.
7. **Source** – Choose your GitHub (Cloud) source details.

   a. **GitHub (Cloud) source** – Choose GitHub (Cloud) Enterprise Cloud.

   b. **GitHub (Cloud) host URL** – Enter the GitHub (Cloud) host name with the protocol (http:// or https://). For example: *`https://api.github.com`*.

   c. **GitHub (Cloud) organization name** – You can find your organization name when you log in to GitHub (Cloud) desktop and go to **Your organizations** under your profile picture dropdown.
8. **Authorization** – Amazon Q Business crawls ACL information by default to ensure responses are generated only from documents your end users have access to. See Authorization for more details.
9. **Authentication** – Enter the following information for your **AWS Secrets Manager secret**.

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

   b. **GitHub (Cloud) token** – Enter the access token you created in GitHub (Cloud).
10. **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.
11. **Identity crawler** – Amazon Q crawls identity information from your data source by default to ensure responses are generated only from documents end users have access to. For more information, see Identity crawler.

12. **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.

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

    a. **Select repositories to crawl**—Select between crawling **All** repositories or **Select repositories**.

       If you choose **Select repositories**, add names for the repositories in **Name of repository** and, optionally, the name of any specific branches in **Name of branch**.

    b. For **Maximum single file size** – Specify the file size limit in MBs that Amazon Q will crawl. Amazon Q will crawl only the files within the size limit you define. The default file size is 50MB. The maximum file size should be greater than 0MB and less than or equal to 50MB.

    c. **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.

14. 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.

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

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

17. **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.

> **Note**
>
> Support for adding custom fields varies by connector. You won't see the **Add field** option if your connector doesn't support adding custom fields.

For more information, see Field mappings.

18. 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.

19. 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 choose to view Amazon 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.
> - You can also view a document-level report in CloudWatch for your data source sync job by selecting **View Report**. This report will have details about the progress and status of each document in the sync job. It shows if a document succeeded, failed, or was skipped during the crawl, sync, and index stages. You'll also find any error messages related to failed or skipped documents.