

Implement web crawling in Amazon Bedrock Knowledge Bases

by Hardik Vasa and Malini Chatterjee | on 30 JUL 2024 | in [Advanced \(300\)](#), [Amazon Bedrock](#), [Amazon Machine Learning](#), [Amazon OpenSearch Service](#), [Amazon Simple Storage Service \(S3\)](#), [Announcements](#), [Best Practices](#), [Technical How-to](#) | [Permalink](#) | [Comments](#) | [Share](#)

[Amazon Bedrock](#) is a fully managed service that offers a choice of high-performing foundation models (FMs) from leading artificial intelligence (AI) companies like AI21 Labs, Anthropic, Cohere, Meta, Stability AI, and Amazon through a single API, along with a broad set of capabilities to build [generative AI](#) applications with security, privacy, and responsible AI.

With Amazon Bedrock, you can experiment with and evaluate top FMs for various use cases. It allows you to privately customize them with your enterprise data using techniques like Retrieval Augmented Generation (RAG), and build agents that run tasks using your enterprise systems and data sources. [Amazon Bedrock Knowledge Bases](#) enables you to aggregate data sources into a repository of information. With knowledge bases, you can effortlessly build an application that takes advantage of RAG.

Accessing up-to-date and comprehensive information from various websites is crucial for many AI applications in order to have accurate and relevant data. Customers using Amazon Bedrock Knowledge Bases want to extend the capability to crawl and index their public-facing websites. By integrating web crawlers into the knowledge base, you can gather and utilize this web data efficiently. In this post, we explore how to achieve this seamlessly.

Web crawler for knowledge bases

With a web crawler data source in the knowledge base, you can create a generative AI web application for your end-users based on the website data you crawl using either the [AWS Management Console](#) or the API. The default crawling behavior of the web connector starts by fetching the provided seed URLs and then traversing all child links within the same top primary domain (TPD) and having the same or deeper URL path.

The current considerations are that the URL can't require any authentication, it can't be an IP address for its host, and its scheme has to start with either `http://` or `https://`. Additionally, the web connector will fetch non-HTML supported files such as PDFs, text files, markdown files, and CSVs referenced in the crawled pages regardless of their URL, as long as they aren't explicitly excluded. If multiple seed URLs are provided, the web connector will crawl a URL if it fits any seed URL's TPD and path. You can have up to 10 source URLs, which the knowledge base uses to as a starting point to crawl.

However, the web connector doesn't traverse pages across different domains by default. The default behavior, however, will retrieve supported non-HTML files. This makes sure the crawling process remains within the specified boundaries, maintaining focus and relevance to the targeted data sources.

Understanding the sync scope

When setting up a knowledge base with web crawl functionality, you can choose from different sync types to control which webpages are included. The following table shows the example paths that will be crawled given the source URL for different sync scopes (`https://example.com` is used for illustration purposes).

Sync Scope Type	Source URL	Example Domain Paths Crawled	Description
Default	<code>https://example.com/products</code>	<code>https://example.com/products</code>	Same host and the

		https://example.com/products/product1 https://example.com/products/product https://example.com/products/discounts	same initial path as the source URL
Host only	https://example.com/sellers	https://example.com/ https://example.com/products https://example.com/sellers https://example.com/delivery	Same host as the source URL
Subdomains	https://example.com	https://blog.example.com https://blog.example.com/posts/post1 https://discovery.example.com https://transport.example.com	Subdomain of the primary domain of the source URLs

You can set the maximum throttling for crawling speed to control the maximum crawl rate. Higher values will reduce the sync time. However, the crawling job will always adhere to the domain's `robots.txt` file if one is present, respecting standard robots.txt directives like 'Allow', 'Disallow', and crawl rate.

You can further refine the scope of URLs to crawl by using inclusion and exclusion filters. These filters are regular expression (regex) patterns applied to each URL. If a URL matches any exclusion filter, it will be ignored. Conversely, if inclusion filters are set, the crawler will only process URLs that match at least one of these filters that are still within the scope. For example, to exclude URLs ending in `.pdf`, you can use the regex `^.*\.pdf$`. To include only URLs containing the word "products," you can use the regex `.*products.*`.

Solution overview

In the following sections, we walk through the steps to create a knowledge base with a web crawler and test it. We also show how to create a knowledge base with a specific embedding model and an [Amazon OpenSearch Service](#) vector collection as a vector database, and discuss how to monitor your web crawler.

Prerequisites

Make sure you have permission to crawl the URLs you intend to use, and adhere to the [Amazon Acceptable Use Policy](#). Also make sure any bot detection features are turned off for those URLs. A web crawler in a knowledge base uses the user-agent `bedrockbot` when crawling webpages.

Create a knowledge base with a web crawler

Complete the following steps to implement a web crawler in your knowledge base:

1. On the Amazon Bedrock console, in the navigation pane, choose **Knowledge bases**.
2. Choose **Create knowledge base**.
3. On the **Provide knowledge base details** page, set up the following configurations:
 - a. Provide a name for your knowledge base.
 - b. In the **IAM permissions** section, select **Create and use a new service role**.
 - c. In the **Choose data source** section, select **Web Crawler** as the data source.
 - d. Choose **Next**.
4. On the **Configure data source** page, set up the following configurations:
 - a. Under **Source URLs**, enter `https://www.aboutamazon.com/news/amazon-offices`.
 - b. For **Sync scope**, select **Host only**.
 - c. For **Include patterns**, enter `^https?://www.aboutamazon.com/news/amazon-offices/.*$`.
 - d. For exclude pattern, enter `.*plants.*` (we don't want any post with a URL containing the word "plants").
 - e. For **Content chunking and parsing**, chose **Default**.
 - f. Choose **Next**.
5. On the **Select embeddings model and configure vector store** page, set up the following configurations:
 - a. In the **Embeddings model** section, chose **Titan Text Embeddings v2**.
 - b. For **Vector dimensions**, enter `1024`.
 - c. For **Vector database**, choose **Quick create a new vector store**.
 - d. Choose **Next**.
6. Review the details and choose **Create knowledge base**.

In the preceding instructions, the combination of **Include patterns** and **Host only** sync scope is used to demonstrate the use of the include pattern for web crawling. The same results can be achieved with the default sync scope, as we learned in the previous section of this post.

Review and create

Step 1: Provide details

Edit

Knowledge base details

Knowledge base name knowledge-base-web-crawl-blog	Knowledge base description —	Service role AmazonBedrockExecutionRoleForKnowledgeBase_4nmrs
---	--	---

Tags (0)

Key	Value
No tags to display	

Step 2: Setup up data source

Edit

Data source: knowledge-base-quick-start-bf8g2-data-source

Data source type WEB	Data source name knowledge-base-quick-start-bf8g2-data-source	Sync scope Host only
Authentication No Authentication		

Step 3: Select embeddings model and configure vector store

Edit

Embeddings model

Model Titan Text Embeddings v2	Vector dimensions 1024
--	----------------------------------

Vector store

Quick create vector store - Recommended
We will create an Amazon OpenSearch Serverless vector store in your account on your behalf.

Cancel

Previous

Create knowledge base

You can use the **Quick create vector store** option when creating the knowledge base to create an [Amazon OpenSearch Serverless](#) vector search collection. With this option, a public vector search collection and vector index is set up for you with the required fields and necessary configurations. Additionally, Amazon Bedrock Knowledge Bases manages the end-to-end ingestion and query workflows.

Test the knowledge base

Let's go over the steps to test the knowledge base with a web crawler as the data source:

1. On the Amazon Bedrock console, navigate to the knowledge base that you created.

<https://aws.amazon.com/blogs/machine-learning/implement-web-crawling-in-knowledge-bases-for-amazon-bedrock/>

4/8

2. Under **Data source**, select the data source name and choose **Sync**. It could take several minutes to hours to sync, depending on the size of your data.
3. When the sync job is complete, in the right panel, under **Test knowledge base**, choose **Select model** and select the model of your choice.
4. Enter one of the following prompts and observe the response from the model:
 - a. How do I tour the Seattle Amazon offices?
 - b. Provide me with some information about Amazon's HQ2.
 - c. What is it like in the Amazon's New York office?

As shown in the following screenshot, citations are returned within the response reference webpages. The value of `x-amz-bedrock-kb-source-uri` is a webpage link, which helps you verify the response accuracy.

Test Knowledge base

Generate responses

Claude 3 Sonnet v1 | ODT

Change

Configure your retrieval and responses

To customize the search strategy for your knowledge base, select the configurations icon .

How do I tour the Seattle Amazon offices?

You can take a self-guided tour of Amazon's Puget Sound headquarters campus in Seattle. The tour allows you to walk through the urban campus and see landmarks like the Spheres while learning about the company and its history.^[1] To take the self-guided tour, you can follow the map provided on the Amazon website that highlights points of interest around the Seattle headquarters. No reservation is required, you can simply show up and explore the public areas of the campus at your own pace.^[2]

< Hide source details

Source details (2)

Each container below shows a chunk that comes from a data source and that relates to your query.

Query configurations (1)

Source chunk 1

How to tour Seattle offices at Amazon's headquarters 1. News 2. Amazon offices 2 min April 17, 2024 ----- Written by Amazon Staff -----
 ----- Facebook](https://www.facebook.com/dialog/share?app_id=2194788737494754&display=popup&href=)[Twitter](https://twitter.com/intent/tweet?url=)[LinkedIn](https://www.linkedin.com/shareArticle?url=&mini=true&title=&summary=&source=US%20About%20Amazon)[Mail To Link copied # How to tour Amazon's Puget Sound headquarters on your next trip to Seattle Written by Amazon Staff An aerial photo of the spheres at Amazon's headquarters in Seattle. Take a self-guided tour of the Seattle campus and landmarks at our headquarters. A map of the Amazon Puget Sound Headquarters Tour.Amazon Puget Sound Headquarters Tour map. You can now tour Amazon's Puget Sound headquarters campus. The new, self-guided experience shares information about the company and its history as you walk through its urban Seattle campus and landmarks.

Metadata associated with this chunk

Key	Value
x-amz-bedrock-kb-source-uri	https://www.aboutamazon.com/news,
x-amz-bedrock-kb-chunk-id	1%3A0%3A9vbEnZABn6jFpZKHGA2J
x-amz-bedrock-kb-data-source-id	HIWFRKEQNZ

Source chunk 2

Create a knowledge base using the AWS SDK

This following code uses the [AWS SDK for Python \(Boto3\)](#) to create a knowledge base in Amazon Bedrock with a specific embedding model and OpenSearch Service vector collection as a vector database:

```
import boto3

client = boto3.client('bedrock-agent')

response = client.create_knowledge_base(
    name='workshop-aoss-knowledge-base',
    roleArn='your-role-arn',
    knowledgeBaseConfiguration={
        'type': 'VECTOR',
        'vectorKnowledgeBaseConfiguration': {
            'embeddingModelArn': 'arn:aws:bedrock:your-region::foundation-model/amazon.titan-emb',
        }
    },
    storageConfiguration={
        'type': 'OPENSEARCH_SERVERLESS',
        'opensearchServerlessConfiguration': {
            'collectionArn': 'your-opensearch-collection-arn',
            'vectorIndexName': 'blog_index',
        }
    }
)
```

The following Python code uses Boto3 to create a web crawler data source for an Amazon Bedrock knowledge base, specifying URL seeds, crawling limits, and inclusion and exclusion filters:

```
import boto3

client = boto3.client('bedrock-agent', region_name='us-east-1')

knowledge_base_id = 'knowledge-base-id'

response = client.create_data_source(
    knowledgeBaseId=knowledge_base_id,
    name='example',
    description='test description',
    dataSourceConfiguration={
        'type': 'WEB',
        'webConfiguration': {
            'sourceConfiguration': {
                'urlConfiguration': {
                    'seedUrls': [
                        {'url': 'https://example.com/'}
                    ]
                }
            }
        }
    }
)
```

Monitoring

You can track the status of an ongoing web crawl in your [Amazon CloudWatch](#) logs, which should report the URLs being visited and whether they are successfully retrieved, skipped, or failed. The following screenshot shows the CloudWatch logs for the crawl job.

▶	2024-07-11T21:58:02.694Z	{ "event_timestamp": 1720735082694, "event": { "ingestion_job_id": "Q5LQWM5W0N", "document_location": {
▶	2024-07-11T21:58:02.694Z	{ "event_timestamp": 1720735082694, "event": { "ingestion_job_id": "Q5LQWM5W0N", "document_location": {
▼	2024-07-11T21:58:02.694Z	{ "event_timestamp": 1720735082694, "event": { "ingestion_job_id": "Q5LQWM5W0N", "document_location": {
		<pre> { "event_timestamp": 1720735082694, "event": { "ingestion_job_id": "Q5LQWM5W0N", "document_location": { "type": "WEB", "web_location": { "url": "https://www.aboutamazon.com/news/amazon-offices/amazon-opens-hank-office-at-lord-and-taylor-building-in-nyc" } } }, "data_source_id": "BFDBEF0VGT", "status_reasons": [], "knowledge_base_arn": "arn:aws:bedrock:us-east-1:[REDACTED]:knowledge-base/KASH8A0IJP", "status": "SCHEDULED_FOR_INGESTION" }, "event_version": "1.0", "event_type": "StartIngestionJob.ResourceStatusChanged", "level": "INFO" } </pre>
▼	2024-07-11T21:58:02.694Z	{ "event_timestamp": 1720735082694, "event": { "ingestion_job_id": "Q5LQWM5W0N", "data_source_id": "BFDB...
		<pre> { "event_timestamp": 1720735082694, "event": { "ingestion_job_id": "Q5LQWM5W0N", "data_source_id": "BFDBEF0VGT", "ingestion_job_status": "CRAWLING_COMPLETED", "knowledge_base_arn": "arn:aws:bedrock:us-east-1:[REDACTED]:knowledge-base/KASH8A0IJP", "resource_statistics": { "number_of_resources_updated": 0, "number_of_resources_ingested": 0, "number_of_resources_scheduled_for_update": 0, "number_of_resources_scheduled_for_ingestion": 13, "number_of_resources_scheduled_for_metadata_update": 0, "number_of_resources_deleted": 0, "number_of_resources_with_metadata_updated": 0, "number_of_resources_failed": 0, "number_of_resources_scheduled_for_deletion": 0 } }, "event_version": "1.0", "event_type": "StartIngestionJob.StatusChanged", "level": "INFO" } </pre>

Clean up

To clean up your resources, complete the following steps:

1. Delete the knowledge base:
 - a. On the Amazon Bedrock console, choose **Knowledge bases** under **Orchestration** in the navigation pane.
 - b. Choose the knowledge base you created.
 - c. Take note of the [AWS Identity and Access Management](#) (IAM) service role name in the knowledge base overview.
 - d. In the **Vector database section**, take note of the OpenSearch Serverless collection ARN.
 - e. Choose **Delete**, then enter `delete` to confirm.
2. Delete the vector database:
 - a. On the OpenSearch Service console, choose **Collections** under **Serverless** in the navigation pane.
 - b. Enter the collection ARN you saved in the search bar.

- c. Select the collection and chose **Delete**.
- d. Enter `confirm` in the confirmation prompt, then choose **Delete**.

3. Delete the IAM service role:

- a. On the IAM console, choose **Roles** in the navigation pane.
- b. Search for the role name you noted earlier.
- c. Select the role and choose **Delete**.
- d. Enter the role name in the confirmation prompt and delete the role.

Conclusion

In this post, we showcased how Amazon Bedrock Knowledge Bases now supports the web data source, enabling you to index public webpages. This feature allows you to efficiently crawl and index websites, so your knowledge base includes diverse and relevant information from the web. By taking advantage of the infrastructure of Amazon Bedrock, you can enhance the accuracy and effectiveness of your generative AI applications with up-to-date and comprehensive data.

For pricing information, see [Amazon Bedrock pricing](#). To get started using Amazon Bedrock Knowledge Bases, refer to [Create a knowledge base](#). For deep-dive technical content, refer to [Crawl web pages for your Amazon Bedrock knowledge base](#). To learn how our Builder communities are using Amazon Bedrock in their solutions, visit our [community.aws](#) website.

About the Authors



Hardik Vasa is a Senior Solutions Architect at AWS. He focuses on Generative AI and Serverless technologies, helping customers make the best use of AWS services. Hardik shares his knowledge at various conferences and workshops. In his free time, he enjoys learning about new tech, playing video games, and spending time with his family.



Malini Chatterjee is a Senior Solutions Architect at AWS. She provides guidance to AWS customers on their workloads across a variety of AWS technologies. She brings a breadth of expertise in Data Analytics and Machine Learning. Prior to joining AWS, she was architecting data solutions in financial industries. She is very passionate about semi-classical dancing and performs in community events. She loves traveling and spending time with her family.



Like



Share

Comments

Log in to comment