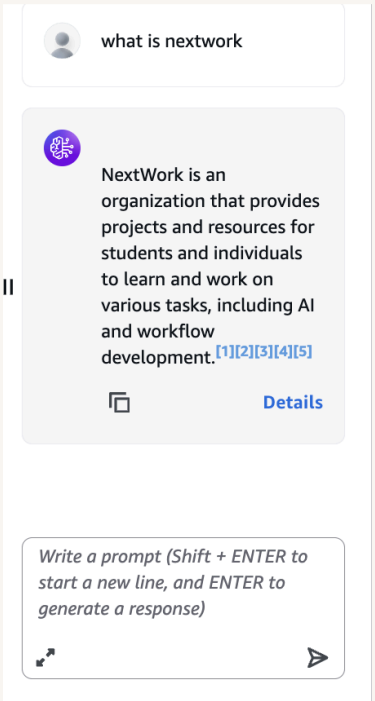
nextwork.org

Set Up a RAG Chatbot in Bedrock

Vinay Pal Singh



Introducing Today's Project!

Start your answer with 'RAG (Retrieval Augmented Generation) is an AI technique that lets you train an AI model on your own personal documents. In this project, We will demonstrate RAG by setting up a RAG chatbot in Amazon Bedrock.

Tools and concepts

Services we used in this project were Amazon Bedrock, S3, and OpenSearch Serverless. Key concepts we learnt include Knowledge Bases, requesting access to AI models, how chatbots generate responses (i.e. AI models + Knowledge base), vector stores.

Project reflection

This project took me approximately two hours including project demo time. The most challenging part was the error with our AI models (and understanding on-demand vs pre-provisioned inference). It was most rewarding to our chatbot responses!

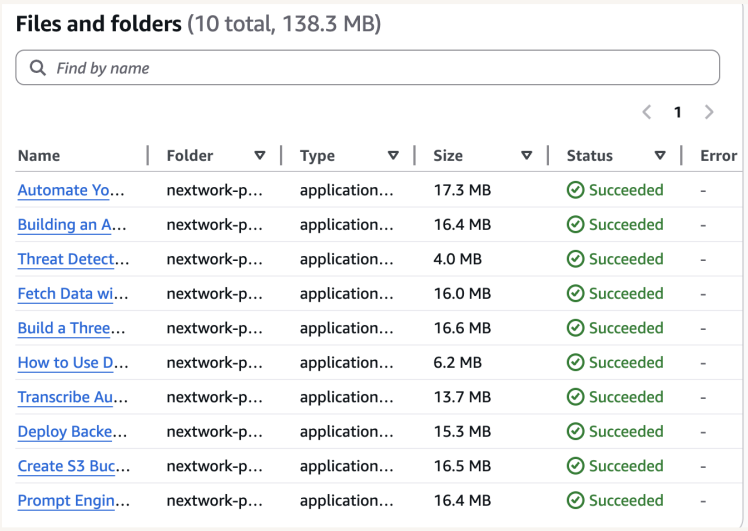
We did this project today to learn more about Bedrock and RAG. This project definetely met our goals - awesome to learn both over a hands on project!

Understanding Amazon Bedrock

Amazon Bedrock is an AWS service that makes it easy to build generative AI applications - It's like an AI model marketplace that lets us find, use and test models from different providers. We're using Bedrock to create a Knowledge Base.

Our Knowledge Base is connected to S3 is going to be the storage/source for our Knowledge Base's raw documents. S3 is AWS's storage service, where you can store all kinds of objects (e.g. videos, documents, audio) in the same bucket.

In an S3 bucket, we uploaded the documents that will make up our chatbot's knowledge. Our S3 bucket is in the same region as our Knowledge Basew because Bedrock is a regional service - data must live in the same region as the Bedrock resource (Kbase)

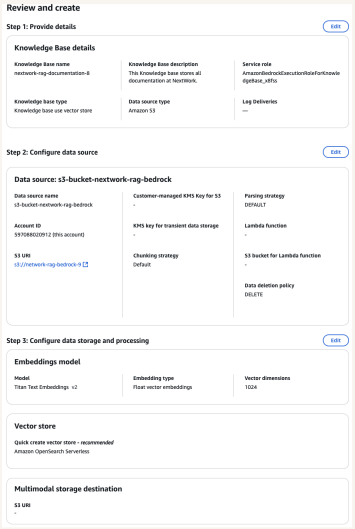

Files and folders (10 total, 138.3 MB)
Find by name
1
Name
Folder
Type
Size
Status
Error
Automate Yo...
nextwork-p...
application...
17.3 MB
Succeeded
-
Building an A...
nextwork-p...
application...
16.4 MB
Succeeded
-
Threat Detect...
nextwork-p...
application...
4.0 MB
Succeeded
-
Fetch Data wi...
nextwork-p...
application...
16.0 MB
Succeeded
-
Build a Three...
nextwork-p...
application...
16.6 MB
Succeeded
-
How to Use D...
nextwork-p...
application...
6.2 MB
Succeeded
-
Transcribe Au...
nextwork-p...
application...
13.7 MB
Succeeded
-
Deploy Backe...
nextwork-p...
application...
15.3 MB
Succeeded
-
Create S3 Buc...
nextwork-p...
application...
16.5 MB
Succeeded
-
Prompt Engin...
nextwork-p...
application...
16.4 MB
Succeeded
-

My Knowledge Base Setup

My Knowledge Base uses a vector store, which means a search engine/database that stores data based on their semantc meaning! When we query our Knowledge Base, OpenSearch will find the relevant chunks of data to the query, and passed it to Bedrock.

Embeddings are vector representations of the semantic meaning of a chunk of test. The embedding model we're using is Titan Text Embeddings v2 because it's fast, accurate and a lot more affordable!

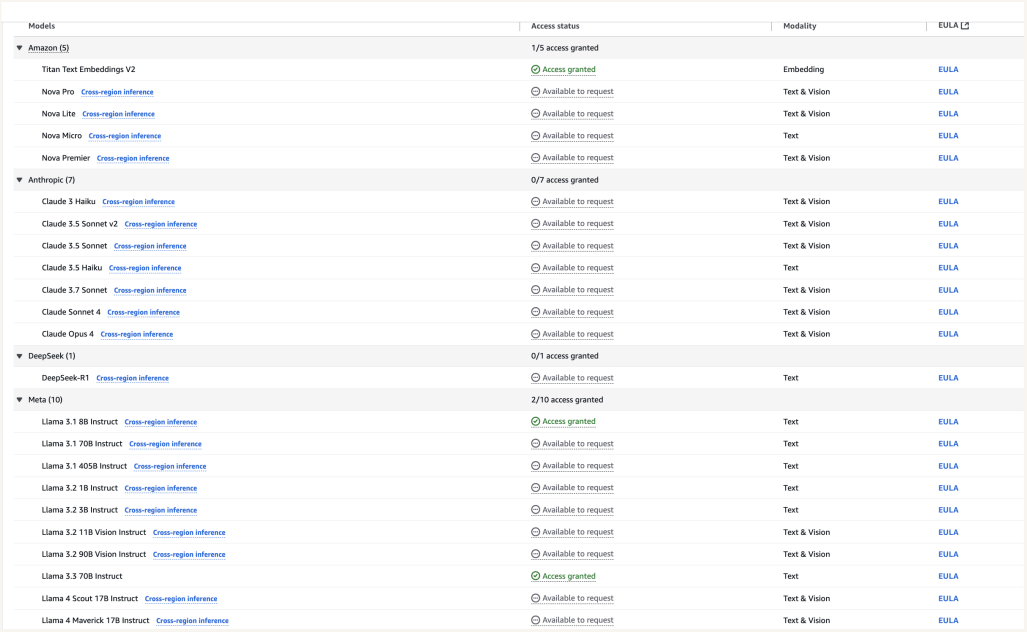
Chunking is the process of splitting up text into smaller pieces i.e. chunks. In our Knowledge Base, chunks are set to be about 300 tokens in size each! chunking helps with searching for data more efficiently in the vector.


Review and create
Step 1: Provide details
Edit
Knowledge Base details
Knowledge Base name
nextwork-rag-documentation-8
Knowledge Base description
This Knowledge base stores all documentation at NextWork.
Service role
AmazonBedrockExecutionRoleForKnowledgeBase_x8fss
Knowledge base type
Knowledge base use vector store
Data source type
Amazon S3
Log Deliveries
—
Step 2: Configure data source
Edit
Data source: s3-bucket-nextwork-rag-bedrock
Data source name
s3-bucket-nextwork-rag-bedrock
Customer-managed KMS Key for S3
-
Parsing strategy
DEFAULT
Account ID
597088002912 (this account)
KMS key for transient data storage
-
Lambda function
-
S3 URI
s3://network-rag-bedrock-9
Chunking strategy
Default
S3 bucket for Lambda function
-
Data deletion policy
DELETE
Step 3: Configure data storage and processing
Edit
Embeddings model
Model
Titan Text Embeddings v2
Embedding type
Float vector embeddings
Vector dimensions
1024
Vector store
Quick create vector store - recommended
Amazon OpenSearch Serverless
Multimodal storage destination
S3 URI
-

AI Models

AI models are important for my chatbot because they're the translator of my Knowledge Base's search resuts into human-like text. Without AI models, our chatbot would only respond with chunks of text from our documents - not the best experience!

To access an AI model in Bedrock, we had to visit the "Model Access" page and request access explicitly! AWS needs explicity access because some AI model providers have extra forms/rules if you wanted to use them, and AWS needs to check availabilty.



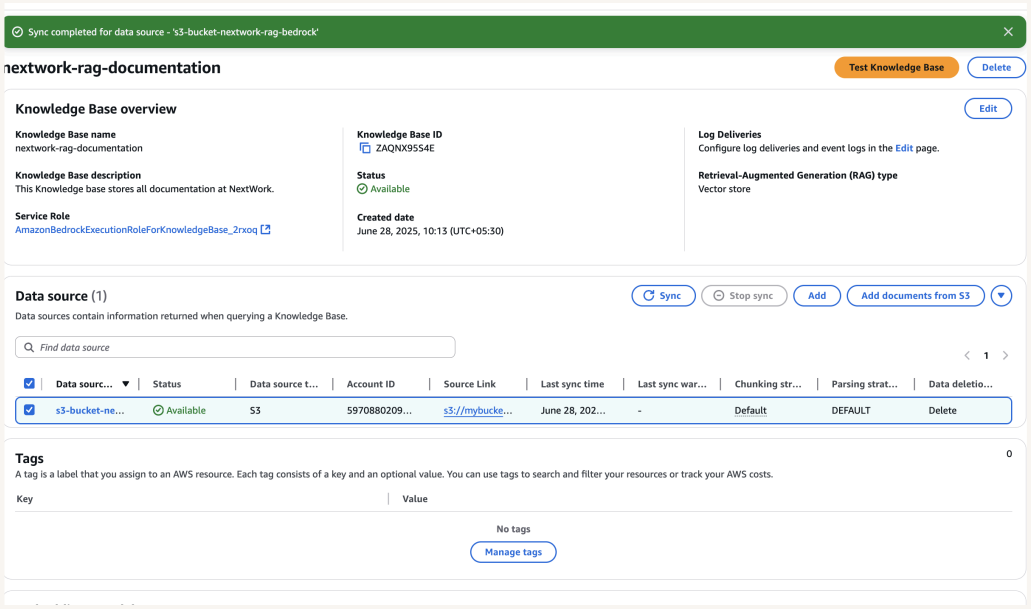
Models	Access status	Modality	EULA
Amazon (5)	1/5 access granted		
Titan Text Embeddings V2	Access granted	Embedding	EULA
Nova Pro Cross-region inference	Available to request	Text & Vision	EULA
Nova Lite Cross-region inference	Available to request	Text & Vision	EULA
Nova Micro Cross-region inference	Available to request	Text	EULA
Nova Premier Cross-region inference	Available to request	Text & Vision	EULA
Anthropic (7)	0/7 access granted		
Claude 3 Haiku Cross-region inference	Available to request	Text & Vision	EULA
Claude 3.5 Sonnet v2 Cross-region inference	Available to request	Text & Vision	EULA
Claude 3.5 Sonnet Cross-region inference	Available to request	Text & Vision	EULA
Claude 3.5 Haiku Cross-region inference	Available to request	Text	EULA
Claude 3.7 Sonnet Cross-region inference	Available to request	Text & Vision	EULA
Claude Sonnet 4 Cross-region inference	Available to request	Text & Vision	EULA
Claude Opus 4 Cross-region inference	Available to request	Text & Vision	EULA
DeepSeek (1)	0/1 access granted		
DeepSeek-R1 Cross-region inference	Available to request	Text	EULA
Meta (10)	2/10 access granted		
Llama 3.1 8B Instruct Cross-region inference	Access granted	Text	EULA
Llama 3.1 70B Instruct Cross-region inference	Available to request	Text	EULA
Llama 3.1 405B Instruct Cross-region inference	Available to request	Text	EULA
Llama 3.2 1B Instruct Cross-region inference	Available to request	Text	EULA
Llama 3.2 3B Instruct Cross-region inference	Available to request	Text	EULA
Llama 3.2 11B Vision Instruct Cross-region inference	Available to request	Text & Vision	EULA
Llama 3.2 90B Vision Instruct Cross-region inference	Available to request	Text & Vision	EULA
Llama 3.3 70B Instruct	Access granted	Text	EULA
Llama 4 Scout 17B Instruct Cross-region inference	Available to request	Text & Vision	EULA
Llama 4 Maverick 17B Instruct Cross-region inference	Available to request	Text & Vision	EULA

Vinay Pal Singh
NextWork Student

Syncing the Knowledge Base

Even through we've already connnected our S3 bucket when creating the Knowledge Base, we still need to sync our data. This is because syncing is what actually moves the data from S3 and into our Knowledge Base + OpenSearch Severless.

The sync process involves three key steps: Ingestion (i.e. Bedrock takes the data from S3), Processing (i.e. Bedrock chunks and embeds the data) and Storing (i.e. Bedrock stores the processed data in the vector store, OpenSearch Serverless).

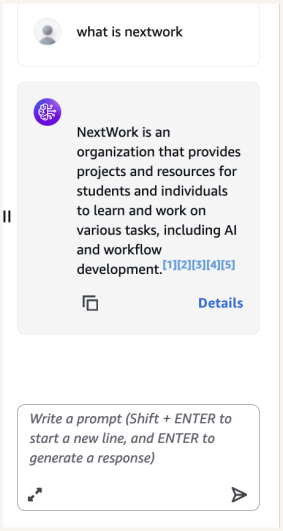


Testing My Chatbot

We initially tried to test our chatbot using Liama 3.1 8B as the AI model, but it triggered an error - it was not available on demand. We had to switch to Liama 3.3 70B because it was offered on-demand by AWS (since it's a newer, efficient model).

When I asked about topics unrelated to our Knowledge Base's data, our chatbot could not help us with the request. This proves that the chatbot only knows the information that we give it - it won't know anything that's outside of our Knowledge Base.

You can also turn off the Generate Responses setting to see the raw chunks of data directly from our Knowledge Base. When we tested this, our chatbot gave a list of 5 paragraphs to answer a question, whereas the AI model will convert it to a sentence


what is nextwork
NextWork is an organization that provides projects and resources for students and individuals to learn and work on various tasks, including AI and workflow development.[1][2][3][4][5]
Details
Write a prompt (Shift + ENTER to start a new line, and ENTER to generate a response)

The place to learn & showcase your skills

Check out nextwork.org for more projects