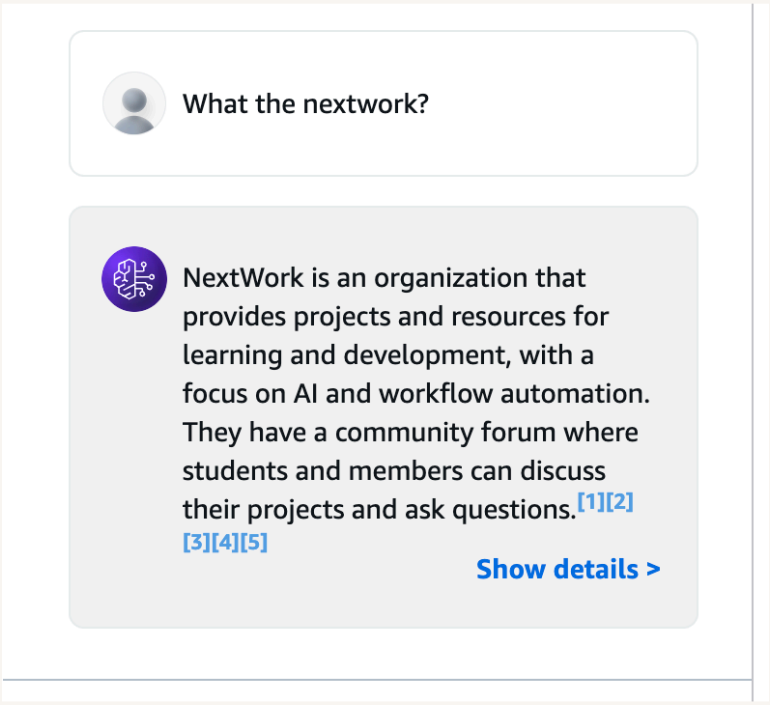
nextwork.org

Set Up a RAG Chatbot in Bedrock

PA pavankumargaggera05@gmail.com



Introducing Today's Project!

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

Tools and concepts

Services I used in this project were Amazon Bedrock, S3 and OpenSearch Serverless. Key concepts I learnt include Knowledge bases, requesting access to AI models, and how chatbot generates responses and vector stores.

Project reflection

This project took me approximately one hour including project demo time.The most challenging part was the error with my AI model. It was most rewarding to level up my chatbot's responses

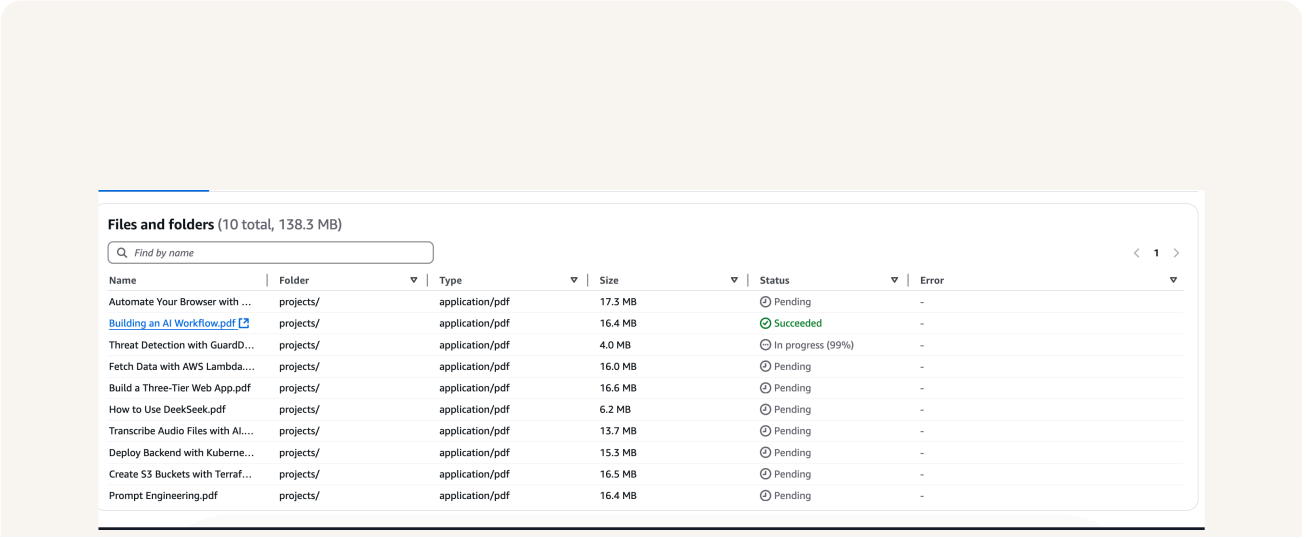
I did this project today to learn more about Bedrock and RAG. This project definitely met my goals - awesome to learn both over a hands on project

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.

My knowledge Base is connected to S3 because S3 is going to be the storage/source for our knowledge Base's raw documents. S3 is AWS's storage service, where we can store all kinds of objects (eg: video, audio, documents) in the same bucket.

In an S3 bucket, I uploaded the documents that will make our chatbot's knowledge. Our S3 bucket is in the same region as our knowledge base because Bedrock is a regional service; data must live in the same region as the bedrock resource (Kbase).

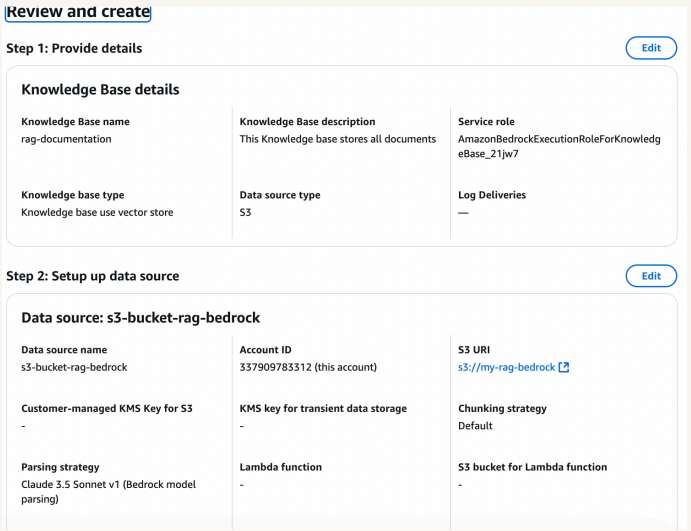


My Knowledge Base Setup

My Knowledge Base uses a vector store, which means a search engine/database that stores data based on their semantic 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 text. The embedding model we are using is Titan text embeddings v2 because it's fast, accurate and a lot more affordable!

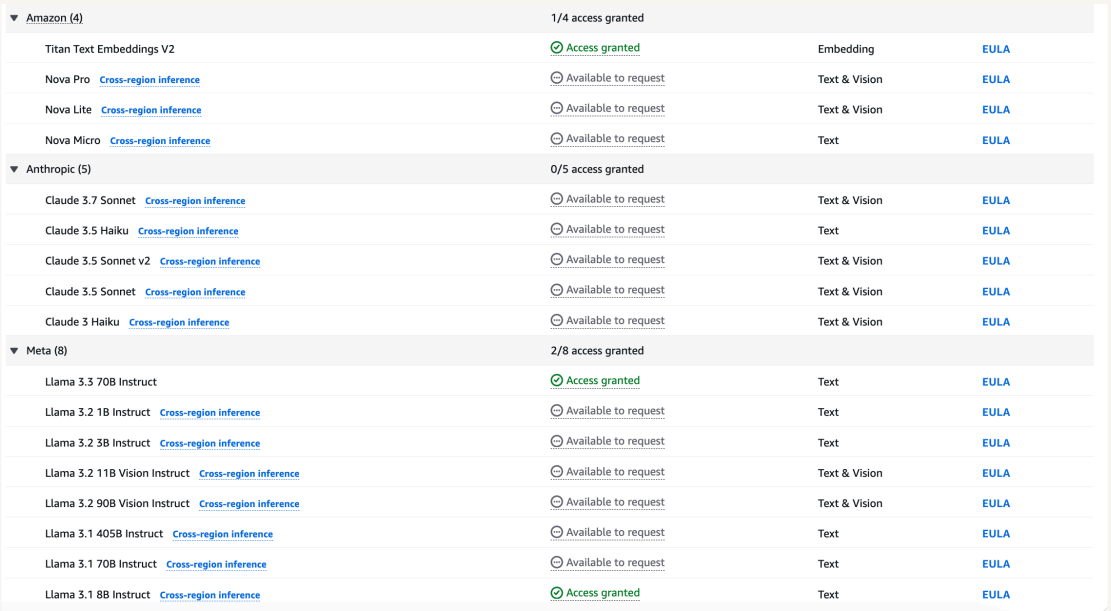
Chunking is the process of splitting the text into smaller pieces, i.e., chunks. In my knowledge base, chunks are set to be about 300 tokens in size each! This helps with searching for data more efficiently in the vector store.



AI Models

AI models are important for my chatbot because they enable it to understand natural language, generate human-like responses, and improve interactions over time. Without AI models, my chatbot would only follow rigid, rule-based scripts.

To get access to AI models in Bedrock, we had to visit the "Model Acess" page and request access explicitly! AWS needs explicit access because some AI model providers have extra forms/rules if you want to use them, and AWS needs to check availability

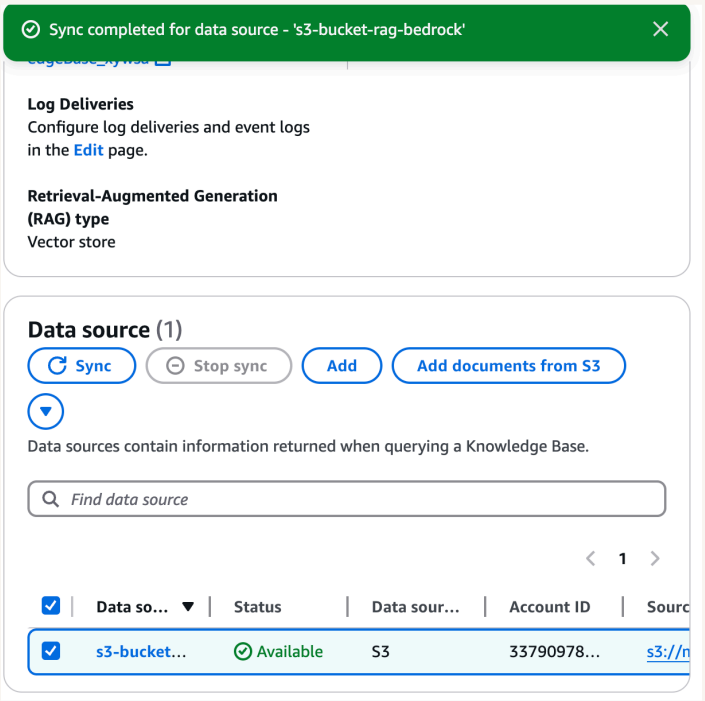


▼ Amazon (4)	1/4 access granted		
Titan Text Embeddings V2	Access granted	Embedding	EULA
Nova Pro Cross-region inference	Available to request	Text & Vision	EULA
Nova Lite Cross-region inference	Available to request	Text & Vision	EULA
Nova Micro Cross-region inference	Available to request	Text	EULA
▼ Anthropic (5)	0/5 access granted		
Claude 3.7 Sonnet Cross-region inference	Available to request	Text & Vision	EULA
Claude 3.5 Haiku Cross-region inference	Available to request	Text	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 Haiku Cross-region inference	Available to request	Text & Vision	EULA
▼ Meta (8)	2/8 access granted		
Llama 3.3 70B Instruct	Access granted	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 3.1 405B Instruct Cross-region inference	Available to request	Text	EULA
Llama 3.1 70B Instruct Cross-region inference	Available to request	Text	EULA
Llama 3.1 8B Instruct Cross-region inference	Access granted	Text	EULA

Syncing the Knowledge Base

Even though I already connected my S3 bucket when creating the Knowledge Base, I still need to sync because Syncing is what actually moves the data from S3 into my knowledge Base + OpenSearch Serverless

The sync process involves three steps: Ingesting (Bedrock takes the data from S3), processing (Bedrock chunks and embeds the data) and storing (Bedrock stores the processed data in the vector store, OpenSearch Serverless).

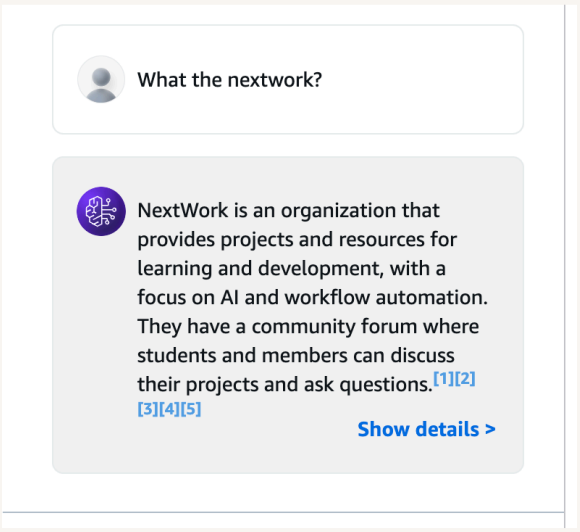


Testing My Chatbot

I initially tried to test my chatbot using Llama 3.1 8B as the AI model, but it triggered an error - it was not available on demand. I had to switch to Llama 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 my data, my chatbot responds that it could not help us with this request. This proves that the chatbot only knows the information we give it - it won't know anything that's outside our Kbase!

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



Everyone should be in a job they love.

Check out nextwork.org for more projects