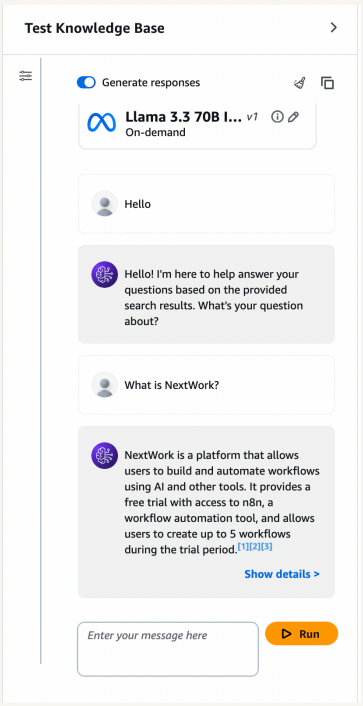
Set Up a RAG Chatbot in Bedrock

Teenamsh Kanamarlapudi



Teenamsh Kanamarlapudi
NextWork Student

Introducing Today's Project!

RAG (Retrieval Augmented Generation) is an AI framework that works like LLMs by allowing them to access external data sources during the generation process. In this project, I will demonstrate RAG by making my bot answer my queries.

Tools and concepts

Services I used were Knowledge Base, S3 Bucket, and Vector Store. Key concepts I learned include embedding and key connections for all services and why they need to be connected.

Project reflection

This project took me approximately 3 hours. The most challenging part was getting my IAM admin user set up because I hadn't had one before to use, so I had to change permissions multiple times. It was most rewarding to get the result.

I did this project to learn about RAG and learn to make my own chatbot. This project definitely met my goal of learning RAG. This was definitely a fun project to work on and do.

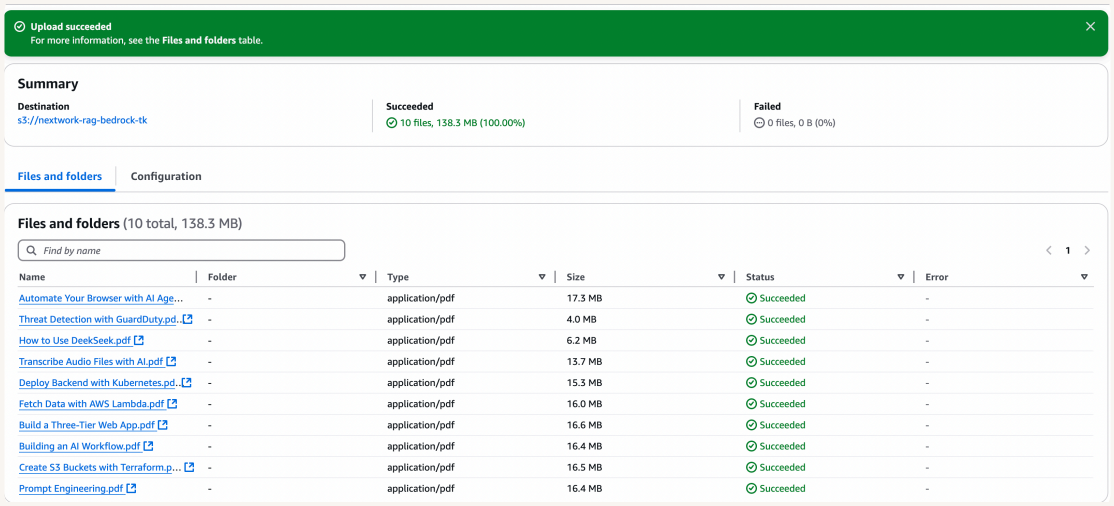
Teenamsh Kanamarlapudi
NextWork Student

Understanding Amazon Bedrock

Amazon Bedrock is an AI model marketplace that allows users to search and use different types of AIs. I'm using Bedrock in this project to use Knowledge Base to create my chatbot.

My Knowledge Base is connected to S3 because that is where my documents will be stored. S3 is a storage system for storing documents, videos, and pictures.

In an S3 bucket, I uploaded 10 different documents for my chatbot. My S3 bucket is in the same region as my Knowledge Base because Bedrock is a region-specific service. Adding on reduces latency.



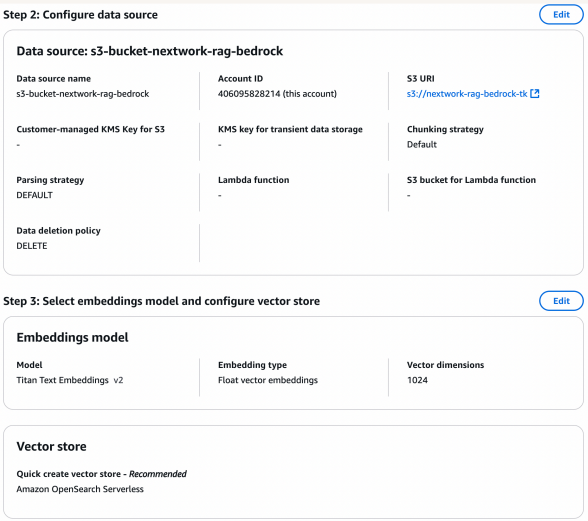
Teenamsh Kanamarlapudi
NextWork Student

My Knowledge Base Setup

My Knowledge Base uses a vector store, which means the AI will find the documents' embeddings to pair up the query and document data. When I query my Knowledge Base, OpenSearch will grab the most relevant chunks of text to answer my question.

Embeddings are special cards that give key phrases, so when a question is asked, it makes another card with the inquiry and connects them to extract information. The embeddings model I'm using is Titan Text Embeddings v2 because it's efficient by AWS

Chunking is a way AI processes larger texts by making them paraphrased so it is more efficient to process information from my documents. In my Knowledge Base, chunks are set to be paraphrased to 300 tokens or so.



Teenamsh Kanamarlapudi
NextWork Student

AI Models

AI models are important for my chatbot because they will give responses human-like responses. Without AI models, my chatbot would only respond with chunks from the documents.

To get access to AI models in Bedrock, I had to select the models to request access to the models. AWS needs explicit access because some are expensive to use, to check if there is enough capacity, and to check all the rules and conditions.

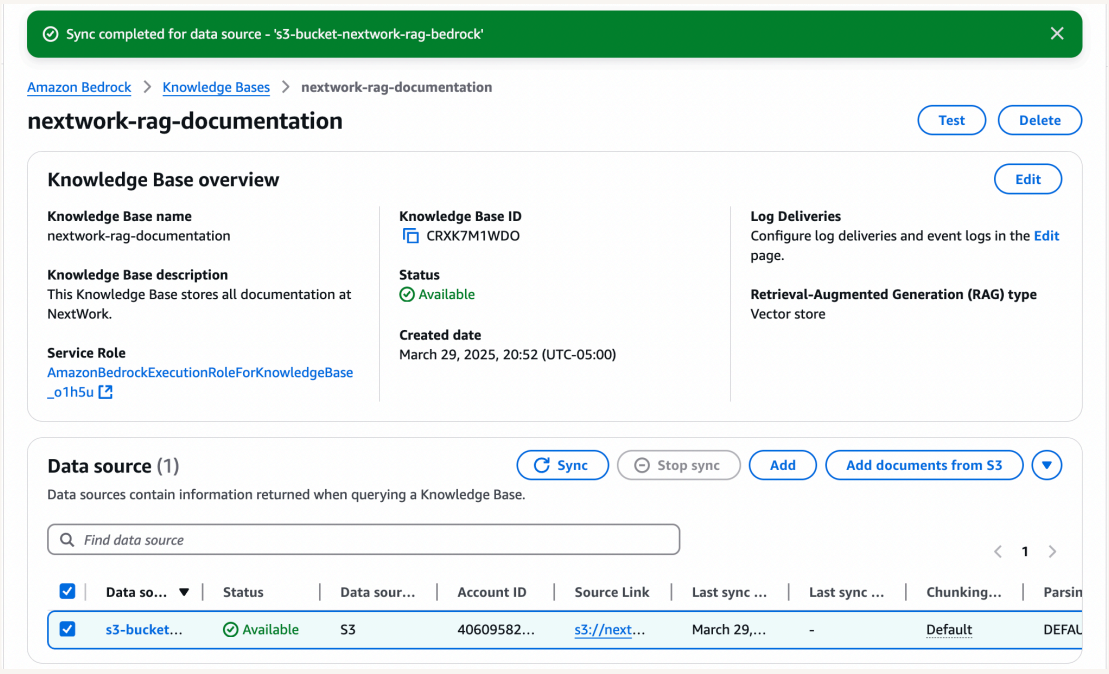
Models	Access status	Modality	EULA
Amazon (4)	1/4 access granted		
Titan Text Embeddings V2	Access granted	Embedding	EULA
Nova Pro Cross-region inference	Available to request	Text & Vision	EULA
Nova Lite Cross-region inference	Available to request	Text & Vision	EULA
Nova Micro Cross-region inference	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
DeepSeek (1)	0/1 access granted		
DeepSeek-R1 Cross-region inference	Available to request	Text	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

Teenamsh Kanamarlapudi
NextWork Student

Syncing the Knowledge Base

Even though I already connected my S3 bucket when creating the Knowledge Base, I still need to sync because the Knowledge Base doesn't have any information inside yet.

The sync process involves three steps: ingesting - receiving data from the data source, processing - chunk and embed the data, and storing - storing in vector store.



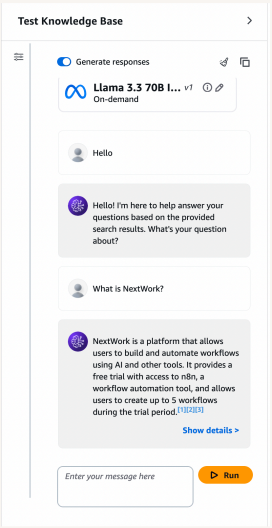
Teenamsh Kanamarlapudi
NextWork Student

Testing My Chatbot

I initially tried to test my chatbot using Llama 3.1 8B as the AI model, but I got an error for my query. I had to switch to Llama 3.3 70B because it is newer and more efficient, they are less resource-intensive.

When I asked about topics unrelated to my data, my chatbot gave me a response saying, "Sorry, I am unable to assist you with this request." This proves that anything outside of the data provided will not be answered.

You can also turn off the Generate Responses setting to get the chunks from the documents to dive deeper into them rather than getting human-like responses.



Everyone should be in a job they love.

Check out nextwork.org for more projects