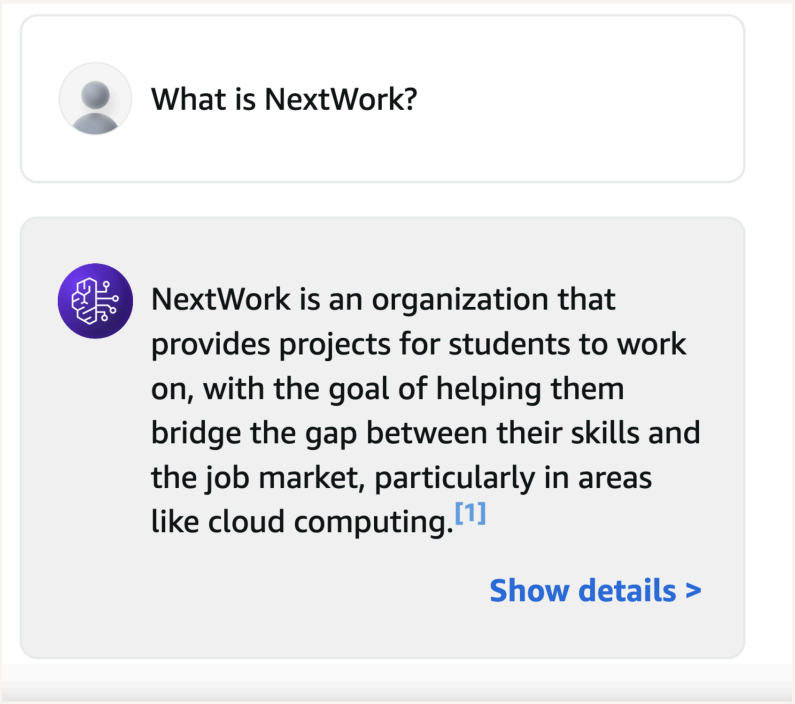
nextwork.org

Set Up a RAG Chatbot in Bedrock

CH chadhaswayam@gmail.com



Introducing Today's Project!

RAG is an AI technique that lets you take an AI model's brain and give it your own documents to train on. In this project, I will demonstrate RAG by creating a chatbot that is trained on my personal documents.

Tools and concepts

Services I used were Knowledge Base in Amazon Bedrock, S3 buckets and vector stores in OpenSearch. Key concepts I learnt include connecting my Knowledge Base to my S3 data Bource and selecting an AI model to chat with my documents.

Project reflection

This project took me approximately an hour. The most challenging part was getting started with creating a Knowledge Base. It was most rewarding to efficiently chat and ask questions to the AI model

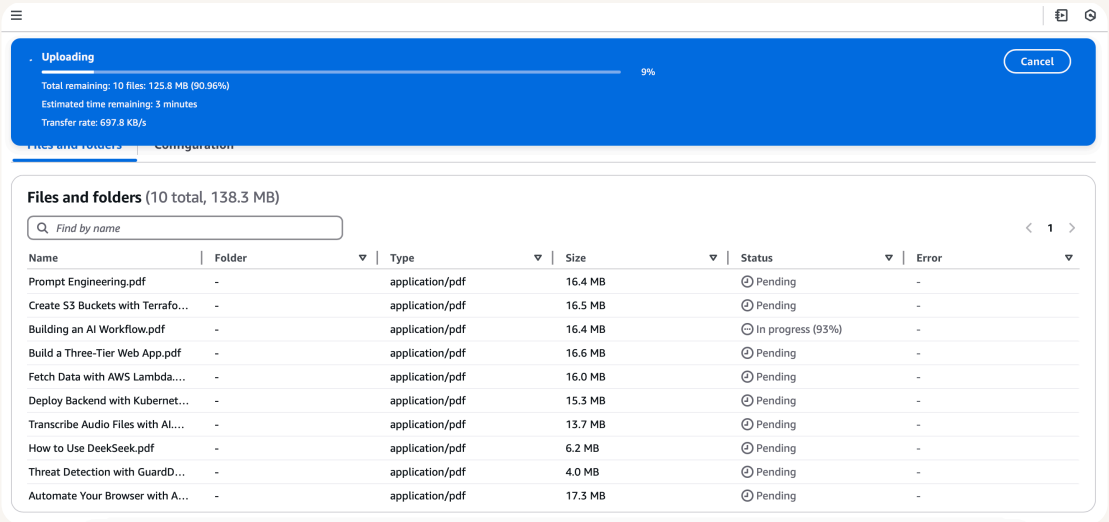
I did this project today to learn more about Amazon Bedrock. It met my goals as it helped me to understand many concepts like Knowledge Bases and OpenSearch

Understanding Amazon Bedrock

Amazon Bedrock is like an AI model marketplace where you can search for and use different models from OpenAI, Meta, etc. I'm using Bedrock in this project to connect a Knowledge Base with an AI model to chat with my documents.

My Knowledge Base is connected to S3 because my documents are stored in a S3 bucket. S3 is an object storage system

In an S3 bucket, I uploaded 10 documents. My S3 bucket is in the same region as my Knowledge Base because Bedrock is a region-specific service, which means your Knowledge Base can only access data in the same region.

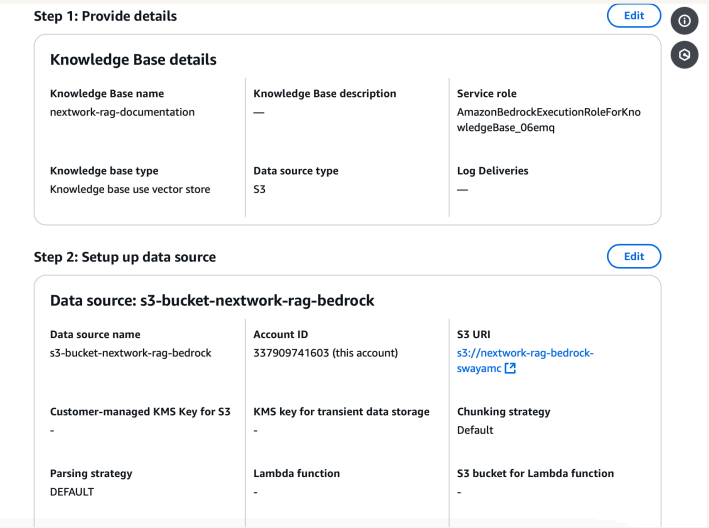


My Knowledge Base Setup

My Knowledge Base uses a vector store, which means a way to store and search for information in a way that understands the meaning of words.When I query my Knowledge Base, OpenSearch will help in searching and visualizing large amounts of data.

Embeddings are representations of values or objects like text, images and audio that are designed to be consumed by machine learning models. The embeddings model I'm using is Titan Text Embeddings v2 because it's fast and works well with AWS services

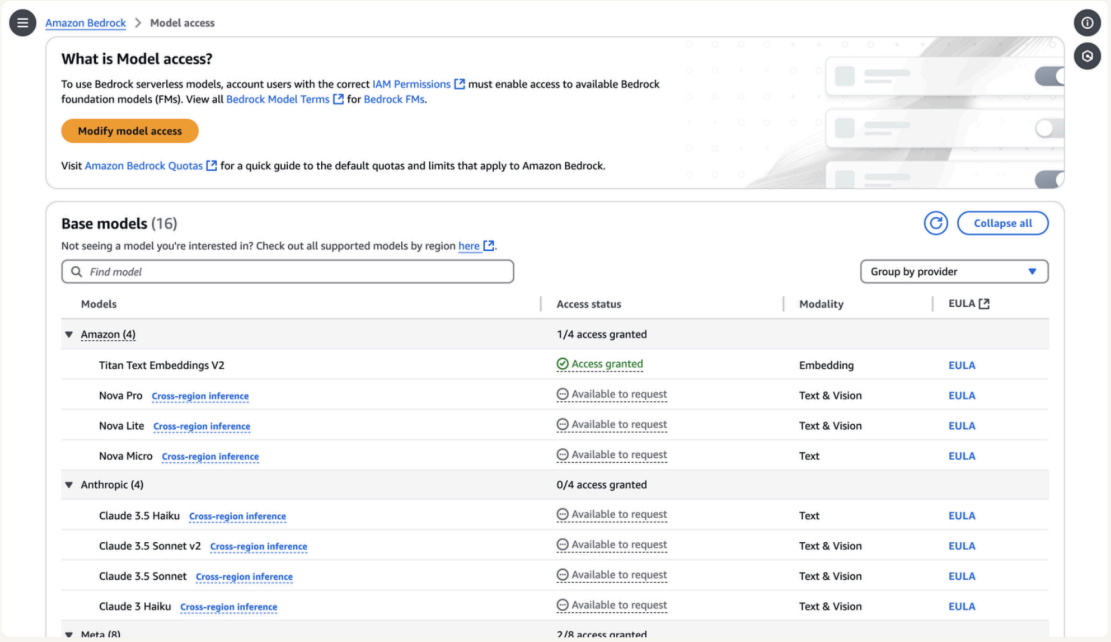
Chunking is breaking a large text into smaller, manageable paragraphs. In my Knowledge Base, chunks are set to 300 tokens.



AI Models

AI models are important for my chatbot because they are the brain behind the chatbot. Without AI models, my chatbot would only respond with chunks of text from my documents

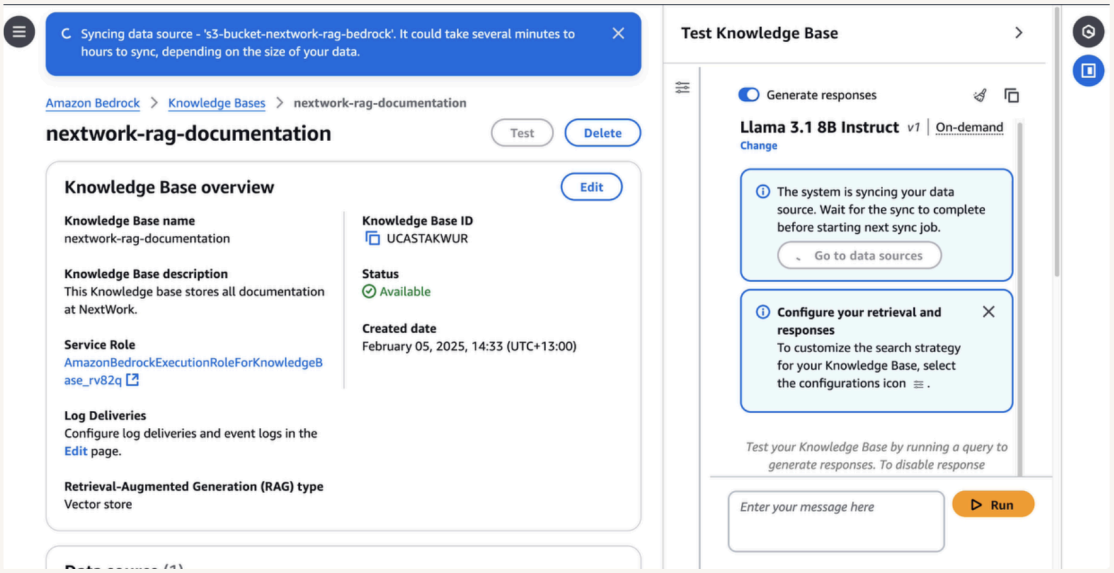
To get access to AI models in Bedrock, I had to explicitly request access from AWS first. AWS needs explicit access because some models are expensive to use, AWS needs to make sure they have enough capacity and some models have different rules.



Syncing the Knowledge Base

Even though I already connected my S3 bucket when creating the Knowledge Base, I still need to sync because the data hasn't actually moved from S3 into my Knowledge Base yet.

The sync process involves three steps: Ingesting(Retrieving the data from the data source), Processing(Chunking and Embedding the data) and Storing(Storing the processed data in the Vector Store)

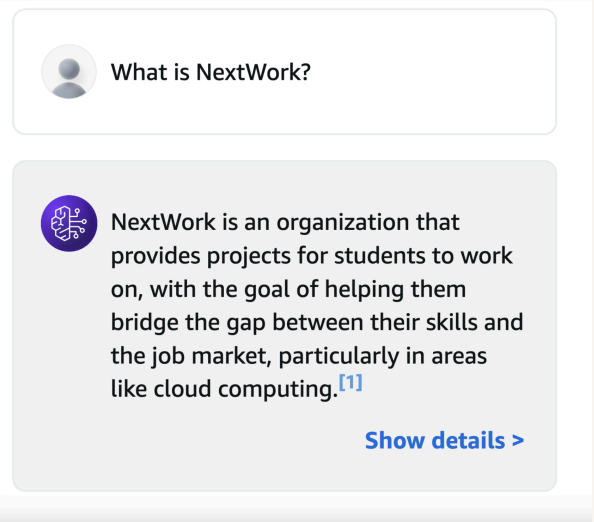


Testing My Chatbot

I initially tried to test my chatbot using Llama 3.1 8B as the AI model, but I got an error. I had to switch to Llama 3.3 70B because it supports on-demand inference.

When I asked about topics unrelated to my data, my chatbot couldn't answer the question. This proves that my chatbot has no knowledge outside of my data.

You can also turn off the Generate Responses setting to just retrieve the processed data directly from your Knowledge Base and analyze it further yourself.



Everyone should be in a job they love.

Check out nextwork.org for more projects