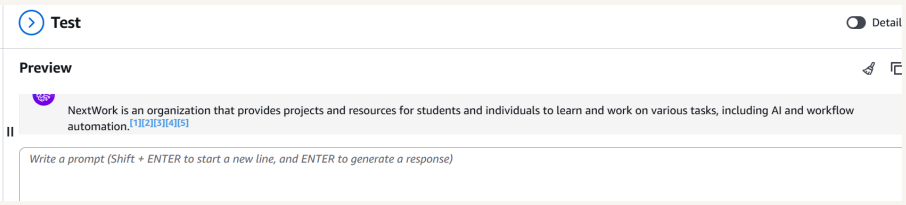
nextwork.org

Set Up a RAG Chatbot in Bedrock

Emmanuel Garrison



Emmanuel Garrison
NextWork Student

Introducing Today's Project!

RAG (Retrieval Augmented Generation) is an AI technique that lets you take an AI model's brain (i.e. their ability to turn data into a human-like response) and give it your own documents to train on. In this project, I will demonstrate RAG by building an AI chatbot that's trained on my personal documents. Such that my AI chatbot becomes an in-house expert on mydocuments and information!

Tools and concepts

Services I used were AWS Bedrock, s3 Bucket and Vector Stores. Key concepts I learnt include Knowledge Bases, AI models, Embeddings, Chunking, RAG, Inference, etc.

Project reflection

This project took me approximately 5 hours.. The most challenging part was setting up the knowledge base. It was most rewarding to build RAG (Retrieval Augmented Generation) chatbot in AWS Bedrock.

I did this project to understand RAG (Retrieval Augmented Generation)

Emmanuel Garrison
NextWork Student

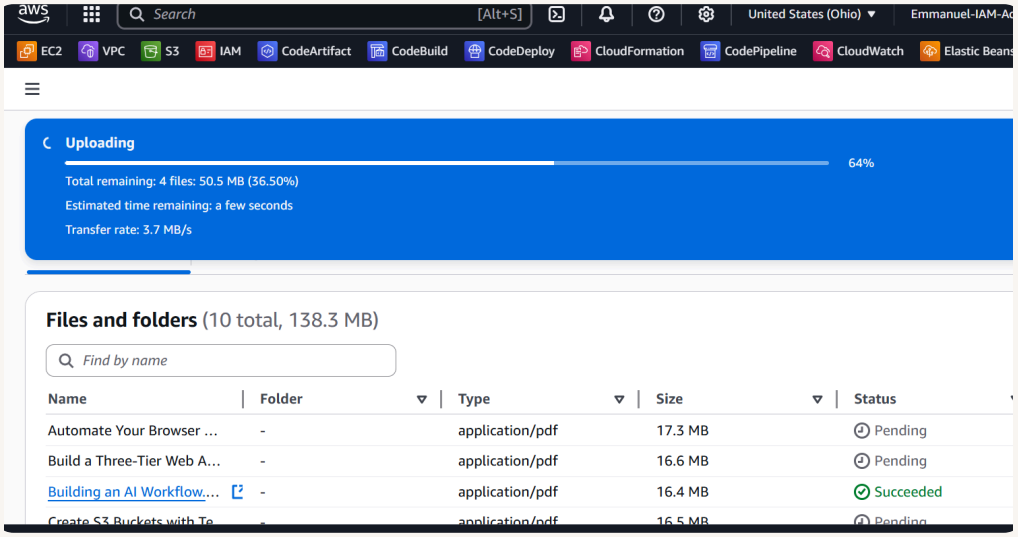
Understanding Amazon Bedrock

Amazon Bedrock is like an AI model marketplace - you can search for and use different models from top companies, like OpenAI, Anthropic, and Meta, in one place. I'm using Bedrock in this project to train AI models on my own information (this process is called RAG). This helps me to create custom models that respond based on my specific data, and is how I'll create a chatbot in this project!

My Knowledge Base is connected to S3 because it's best practice to keep documents in a separate system like S3, so it's possible to manage your documents independently from your chatbot. S3 is a storage system. Think of it as a digital folder where you can store your documents, photos, videos, and other files.

In an S3 bucket, I uploaded 10 files to build my Knowledge Base. 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. In general, keeping your resources in the same region reduces latency (the time it takes for data to travel) and can lower costs too.

Emmanuel Garrison
NextWork Student



Emmanuel Garrison
NextWork Student

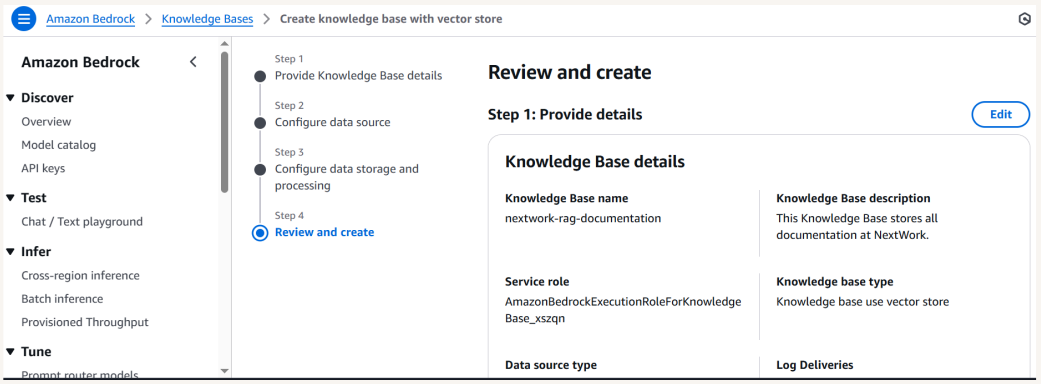
My Knowledge Base Setup

My Knowledge Base uses a vector store, which means a vector store is a way to store and search for information in a way that understands the meaning of words, not just the words themselves. In more technical terms, a vector store is a database that stores your documents' embeddings. In this case, we're using Amazon OpenSearch Serverless as our vector store. OpenSearch will help you search, analyze, and visualize large amounts of data quickly. Once your documents are chunked and embedded, Bedrock will store the embeddings in OpenSearch. Then, when you query your Knowledge Base, Bedrock will go into the OpenSearch vector store to grab the most relevant chunks of text to answer your question.

Embeddings are like a special card or all-purpose tag for each document that lists all its important themes, topics, and content. When you add new documents to your Knowledge Base: The embeddings model reads each piece of text. The embeddings model I'm using is Titan Text Embeddings v2 because it's fast, accurate, and works well with other AWS services, making it a great choice for processing your Knowledge Base documents. Creates a special "card" (actually a list of numbers) that captures what the text is about.

Chunking is breaking a large text into smaller, manageable paragraphs. AI models have a limit on the amount of text they can process at once, so chunking your documents helps the chatbot process information in your Knowledge Base efficiently. In my Knowledge Base, chunks are set to be chunked into 300 tokens, which is like 300 words or punctuation marks.

Emmanuel Garrison
NextWork Student

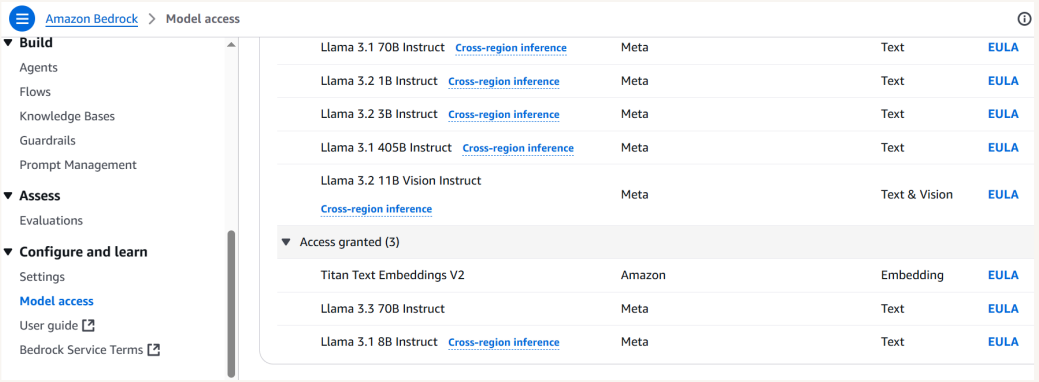


Emmanuel Garrison
NextWork Student

AI Models

AI models are important for my chatbot because it will be the brains behind our chatbot - they'll convert your Knowledge Base's results into human-like text responses. Without AI models, my chatbot would only just respond with chunks of text from your documents (i.e. the raw search results), which isn't the best experience for anyone using the chatbot!

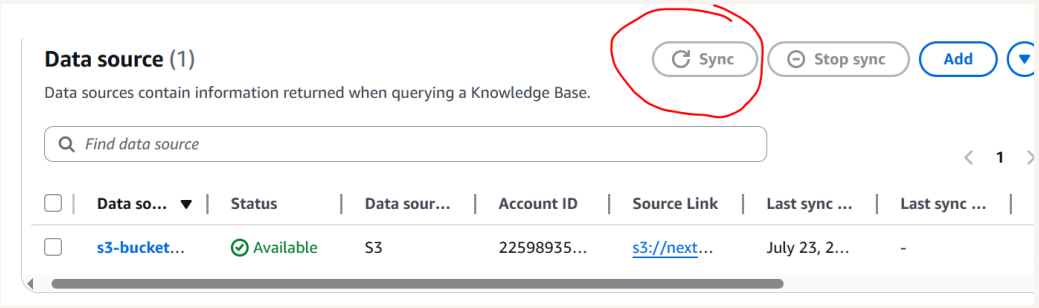
To get access to AI models in Bedrock, I had to navigate to Bedrock configurations, open Model access and select Enable specific models. AWS needs explicit access because... -Some models are expensive to use, so AWS is double checking that you're consciously opting-in to using them. -AWS needs to make sure they have enough capacity i.e. servers in your region to support the model you want to use. -Some models have different rules and conditions that you need to review and accept before you can start using them. For example, Anthropic models need you to fill out a form that tells them what you're going to use the model for....


Amazon Bedrock > Model access
Build
Agents
Flows
Knowledge Bases
Guardrails
Prompt Management
Assess
Evaluations
Configure and learn
Settings
Model access
User guide
Bedrock Service Terms
Llama 3.1 70B Instruct Cross-region inference Meta Text EULA
Llama 3.2 1B Instruct Cross-region inference Meta Text EULA
Llama 3.2 3B Instruct Cross-region inference Meta Text EULA
Llama 3.1 405B Instruct Cross-region inference Meta Text EULA
Llama 3.2 11B Vision Instruct Cross-region inference Meta Text & Vision EULA
Access granted (3)
Titan Text Embeddings V2 Amazon Embedding EULA
Llama 3.3 70B Instruct Meta Text EULA
Llama 3.1 8B Instruct Cross-region inference Meta 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 Creating the Knowledge Base sets up the infrastructure for your chatbot to use. In this case, Bedrock has created a Knowledge Base that's connected to a data source (the S3 bucket) and a vector store (OpenSearch Serverless). But, the data hasn't actually moved from S3 into your Knowledge Base yet.

The sync process involves three steps: -Ingesting - Bedrock will retrieve the data from the data source (S3). -Processing - Bedrock will chunk (i.e. split up into smaller pieces) and embed (i.e. convert into numbers) the data. -Storing - Bedrock will store the processed data in the vector store (OpenSearch Serverless).


Data source (1)
Data sources contain information returned when querying a Knowledge Base.
Sync
Stop sync
Add
Find data source
1
Data so...
Status
Data sour...
Account ID
Source Link
Last sync ...
Last sync ...
s3-bucket...
Available
S3
22598935...
s3://next...
July 23, 2...
-

Emmanuel Garrison
NextWork Student

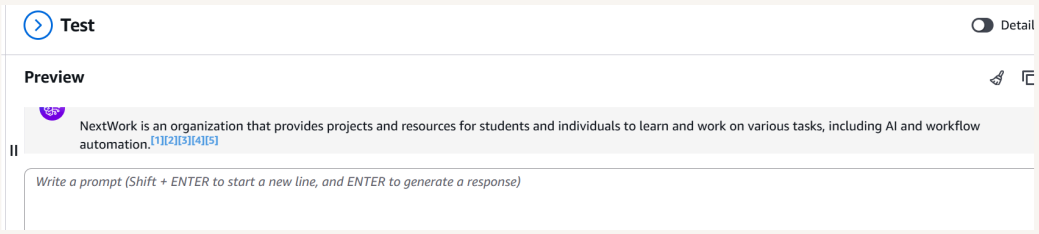
Testing My Chatbot

I initially tried to test my chatbot using Llama 3.1 8B as the AI model, but I got an error "Sorry, I am unable to assist you with this request."...This happens because the AI model I'm using, Llama 3.1 8B, doesn't support on-demand inference! Inference is the process of using an AI model to generate responses. Older or more complex AI models (like Llama 3.1 8B) often need pre-provisioned inference because they need dedicated computing resources to run efficiently. I had to switch to Llama 3.3 70B because it can handle on-demand inference because they're less resource-intensive to start up!

When I asked about topics unrelated to my data, my chatbot says "Sorry, I am unable to assist you with this request." This proves to me that my chatbot has no knowledge outside of my data.

You can also turn off the Generate Responses setting so that your AI model is no longer translating the Knowledge Base's search results into a chat response. This is a great option if you just want to retrieve the processed data directly from your Knowledge Base, to analyze it further yourself.

Emmanuel Garrison
NextWork Student



The place to learn & showcase your skills

Check out nextwork.org for more projects