



[nextwork.org](https://nextwork.org)

# Set Up a RAG Chatbot in Bedrock

CH

chadhaswayam@gmail.com



What is NextWork?



NextWork is an organization that provides projects for students to work on, with the goal of helping them bridge the gap between their skills and the job market, particularly in areas like cloud computing.[\[1\]](#)

[Show details >](#)

# 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 Source 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.



The screenshot shows an S3 upload interface. At the top, a blue progress bar indicates "Uploading" with "9%" completed. Below it, status information includes "Total remaining: 10 files: 125.8 MB (90.96%)", "Estimated time remaining: 3 minutes", and "Transfer rate: 697.8 KB/s". There are tabs for "Files and folders" (which is selected) and "Configuration". The main area displays a table titled "Files and folders (10 total, 138.3 MB)". The table has columns for Name, Folder, Type, Size, Status, and Error. All 10 files listed are PDFs, mostly named "How To ... .pdf", and are currently in a "Pending" status.

Name	Folder	Type	Size	Status	Error
Prompt Engineering.pdf	-	application/pdf	16.4 MB	⌚ Pending	-
Create S3 Buckets with Terrafo...	-	application/pdf	16.5 MB	⌚ Pending	-
Building an AI Workflow.pdf	-	application/pdf	16.4 MB	⌚ In progress (93%)	-
Build a Three-Tier Web App.pdf	-	application/pdf	16.6 MB	⌚ Pending	-
Fetch Data with AWS Lambda....	-	application/pdf	16.0 MB	⌚ Pending	-
Deploy Backend with Kubernetc...	-	application/pdf	15.3 MB	⌚ Pending	-
Transcribe Audio files with AI....	-	application/pdf	13.7 MB	⌚ Pending	-
How to Use DeekSeek.pdf	-	application/pdf	6.2 MB	⌚ Pending	-
Threat Detection with GuardD...	-	application/pdf	4.0 MB	⌚ Pending	-
Automate Your Browser with A...	-	application/pdf	17.3 MB	⌚ Pending	-

# 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.

The screenshot shows the configuration interface for a Knowledge Base. It consists of two main sections: "Step 1: Provide details" and "Step 2: Setup up data source".

**Step 1: Provide details**

Knowledge Base details		
Knowledge Base name nextwork-rag-documentation	Knowledge Base description —	Service role AmazonBedrockExecutionRoleForKnowledgeBase_06emq
Knowledge base type Knowledge base use vector store	Data source type S3	Log Deliveries —

**Step 2: Setup up data source**

Data source: s3-bucket-nextwork-rag-bedrock		
Data source name s3-bucket-nextwork-rag-bedrock	Account ID 337909741603 (this account)	S3 URI <a href="s3://nextwork-rag-bedrock-swayamc">s3://nextwork-rag-bedrock-swayamc</a>
Customer-managed KMS Key for S3 -	KMS key for transient data storage -	Chunking strategy Default
Parsing strategy DEFAULT	Lambda function -	S3 bucket for Lambda function -

# 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.

The screenshot shows the 'Model access' section of the Amazon Bedrock console. At the top, there's a heading 'What is Model access?' with a note about IAM permissions and a 'Modify model access' button. Below this, a link to 'Amazon Bedrock Quotas' is provided. The main area is titled 'Base models (16)' and contains a table with columns for 'Models', 'Access status', 'Modality', and 'EULA'. The table lists models categorized by provider: Amazon (4) and Anthropic (4). Most models have an 'Access granted' status, except for Nova Pro, Nova Lite, and Nova Micro which are 'Available to request'. The table includes a search bar, a 'Group by provider' dropdown, and a 'Collapse all' button. At the bottom, there's a note about Meta (1) and a footer with a '7/8 access granted' message.

Models	Access status	Modality	EULA
Titan Text Embeddings V2	Access granted	Embedding	EULA
Nova Pro <a href="#">Cross-region inference</a>	Available to request	Text & Vision	EULA
Nova Lite <a href="#">Cross-region inference</a>	Available to request	Text & Vision	EULA
Nova Micro <a href="#">Cross-region inference</a>	Available to request	Text	EULA
Claude 3.5 Haiku <a href="#">Cross-region inference</a>	Available to request	Text	EULA
Claude 3.5 Sonnet v2 <a href="#">Cross-region inference</a>	Available to request	Text & Vision	EULA
Claude 3.5 Sonnet <a href="#">Cross-region inference</a>	Available to request	Text & Vision	EULA
Claude 3 Haiku <a href="#">Cross-region inference</a>	Available to request	Text & Vision	EULA

# 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)

The screenshot shows two panels of the Amazon Bedrock interface. On the left, the 'Knowledge Base overview' panel for 'nextwork-rag-documentation' displays basic information: Knowledge Base name (nextwork-rag-documentation), Knowledge Base description (This Knowledge base stores all documentation at NextWork.), Service Role (AmazonBedrockExecutionRoleForKnowledgeBase\_rv82q), Log Deliveries (Configure log deliveries and event logs in the Edit page.), and Retrieval-Augmented Generation (RAG) type (Vector store). A blue banner at the top indicates that the system is syncing data from an S3 bucket. On the right, the 'Test Knowledge Base' panel shows the configuration for a 'Llama 3.1 8B Instruct v1' model set to 'On-demand'. It includes instructions for syncing data and configuring retrieval responses, along with a message input field and a 'Run' button.

# 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.

The screenshot shows a chatbot interface with a light gray background. At the top, there's a white input field containing a user icon and the text "What is NextWork?". Below this, a gray card displays a bot icon (a purple circle with a white gear and brain symbol) and the text: "NextWork is an organization that provides projects for students to work on, with the goal of helping them bridge the gap between their skills and the job market, particularly in areas like cloud computing. [1]". At the bottom right of this card is a blue "Show details >" link. The entire interface is set against a large, rounded rectangular background.



NextWork.org

# **Everyone should be in a job they love.**

Check out nextwork.org for  
more projects

