



## **AWS Project Documentation**

# Set Up a RAG Chatbot in Bedrock

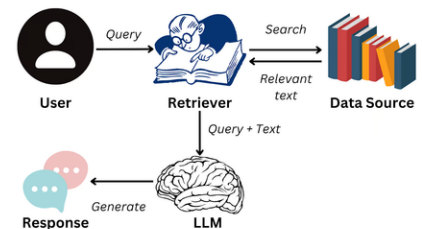


# Notes

## Building a RAG Chatbot in Amazon Bedrock

### Introduction :

Retrieval-Augmented Generation (RAG) is a powerful approach that enhances AI chatbots by integrating an external knowledge base. Instead of relying solely on predefined responses, RAG enables chatbots to retrieve relevant information from stored documents, ensuring accurate and dynamic responses.



### Why Use the RAG Model?

- **Enhanced Accuracy:** Provides fact-based responses by retrieving relevant context.
- **Scalability:** Supports large datasets without increasing computational load.
- **Flexibility:** Adapts to new information without retraining the AI model.



### What is Amazon Bedrock?

Amazon Bedrock is a fully managed AI service that allows developers to build and scale generative AI applications using foundation models (FMs) from various AI providers. It simplifies integrating AI capabilities into applications without requiring infrastructure management.

# Notes

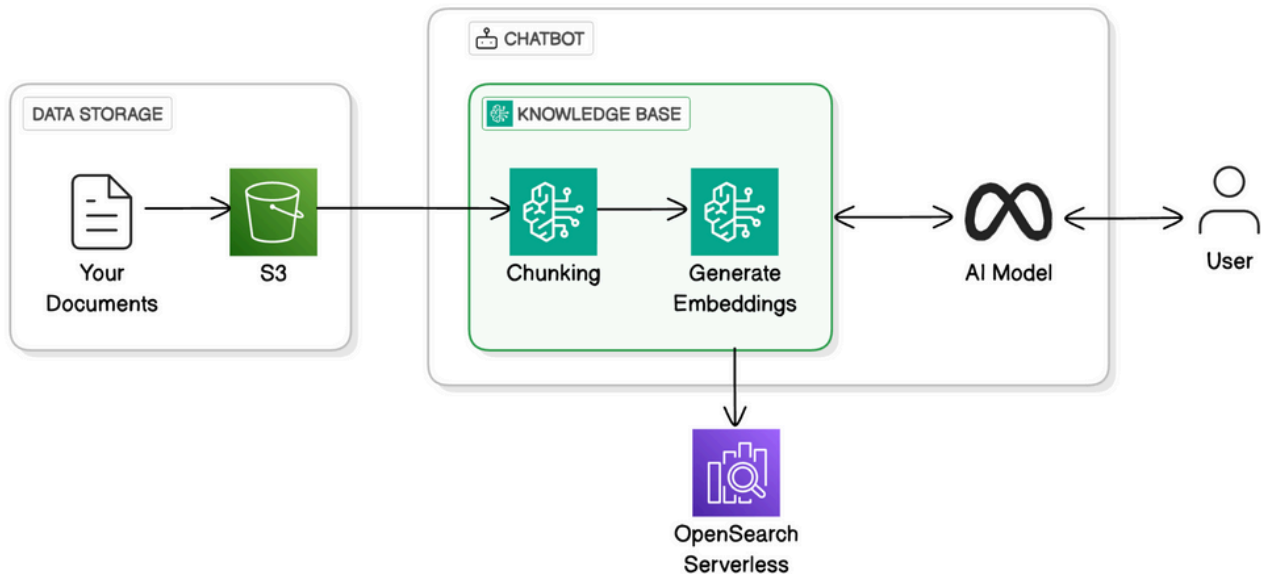
## Why Use Amazon Bedrock?

- **Access to Multiple AI Models:** Supports models from Anthropic, AI21 Labs, Stability AI, and Amazon Titan.
- **Customization:** Fine-tune models with your own data for domain-specific AI.
- **Seamless Integration:** Works with AWS services like S3, OpenSearch, and Lambda.
- **Scalability:** Handles high-volume requests without manual scaling.
- **Security & Compliance:** Built with enterprise-grade security and access controls.

## Use Cases of Amazon Bedrock

- **Chatbots and Virtual Assistants:** Power AI-driven customer support bots.
- **Knowledge Management:** Enhance enterprise search with RAG-based models.
- **Content Generation:** Automate blog writing, reports, and marketing content.
- **Code Assistance:** Improve developer productivity with AI-driven code suggestions.
- **Medical and Legal Research:** Process large text datasets for insights.

## ARCHITECTURE DIAGRAM:



## Architecture Overview :

1. **Data Storage (Amazon S3):** Stores raw documents uploaded by users.
2. **Chunking:** Splits large documents into smaller sections for efficient processing.
3. **Embedding Generation:** Converts chunks into vector representations.
4. **OpenSearch Serverless:** Indexes and retrieves embeddings for quick lookup.
5. **AI Model (Amazon Bedrock):** Processes queries and retrieves relevant data.
6. **User Interaction:** The chatbot interacts with users based on retrieved knowledge.

# Notes

## Steps to Build a RAG Chatbot:

### 1. Store Documents in Amazon S3

- Upload raw documents (PDFs, text files, etc.) to an S3 bucket.
- Configure permissions to allow secure access.

### 2. Chunking the Documents

- Break large documents into smaller, manageable text chunks.
- Use natural language processing (NLP) to ensure meaningful segmentation.

### 3. Generate Embeddings

- Convert each text chunk into a vector representation using an embedding model.
- Store these embeddings for efficient similarity search.

### 4. Index and Search in OpenSearch Serverless

- Store embeddings in Amazon OpenSearch Serverless.
- Use vector search to retrieve the most relevant document chunks for a query.

### 5. AI Model Processing in Amazon Bedrock

- Query the AI model with user input and retrieved document context.
  - Generate an accurate response based on the retrieved knowledge.

# Notes

- Compare different models for response quality:
  - Titan Text Embeddings V2: Converts text into numerical embeddings.
  - Llama 3.1 8B Instruct: Optimized for conversational AI.
  - Llama 3.3 70B Instruct: Advanced reasoning and response generation.

## 6. User Interaction and Refinement

- Deploy the chatbot for real-time user queries.
- Continuously improve response accuracy by refining the knowledge base.
- Experiment with additional models like Claude, Mistral, or Command R+ for better performance.

## Services Used

- Amazon S3: Stores documents securely.
- Amazon OpenSearch Serverless: Enables fast retrieval of relevant information.
- Amazon Bedrock: Hosts AI models for intelligent responses.
- Embedding Model: Converts text into vector representations.

# Notes

## SAMPLE OUTPUT :

Amazon S3 > Buckets > nextwork-rag-bedrock-nivas

nextwork-rag-bedrock-nivas

Objects (10)

Find objects by prefix

Name	Type	Last modified	Size	Storage class
Automate Your Browser with AI Agents.pdf	pdf	March 7, 2025, 18:55:24 (UTC+05:30)	17.3 MB	Standard
Build a Three-Tier Web App.pdf	pdf	March 7, 2025, 18:56:57 (UTC+05:30)	16.6 MB	Standard
Building an AI Workflow.pdf	pdf	March 7, 2025, 19:42:36 (UTC+05:30)	16.4 MB	Standard
Create S3 Buckets with Terraform.pdf	pdf	March 7, 2025, 19:58:34 (UTC+05:30)	16.5 MB	Standard
Deploy Backend with Kubernetes.pdf	pdf	March 7, 2025, 20:04:14 (UTC+05:30)	15.3 MB	Standard
Fetch Data with AWS Lambda.pdf	pdf	March 7, 2025, 20:04:46 (UTC+05:30)	16.0 MB	Standard
How to Use DeepSeek.pdf	pdf	March 7, 2025, 20:04:51 (UTC+05:30)	6.2 MB	Standard
Prompt Engineering.pdf	pdf	March 7, 2025, 20:04:42 (UTC+05:30)	16.4 MB	Standard
Threat Detection with GuardDuty.pdf	pdf	March 7, 2025, 18:40:08 (UTC+05:30)	4.0 MB	Standard
Transcribe Audio Files with AI.pdf	pdf	March 7, 2025, 20:04:55 (UTC+05:30)	13.7 MB	Standard

Amazon Bedrock > Knowledge Bases > Nextwork-rag-documentation

Nextwork-rag-documentation

Knowledge Base overview

Knowledge Base name: Nextwork-rag-documentation

Knowledge Base ID: BTANH0EBXJ

Knowledge Base description: This Knowledge Base stores all documentation at NextWork.

Service Role: AmazonBedrockExecutionRoleForKnowledgeBase\_bpj3v

Status: Available

Created date: March 07, 2025, 20:19 (UTC+05:30)

Log Deliveries: Configure log deliveries and event logs in the [Edit](#) page.

Retrieval-Augmented Generation (RAG) type: Vector store

Data source (1)

Data sources contain information returned when querying a Knowledge Base.

Find data source

Data source	Status	Data source	Account ID	Source Link	Last sync	Last sync	Chunking	Parsing	Data d
s3-bucket...	Available	S3	02609055...	s3://next...	March 07...	-	Default	DEFAULT	Delete

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key: Value

No tags

Manage tags

What AWS services has the student worked on?

1. This project absolutely met our goals and we could see a fully functioning web app by the end. Natasha Ong, NextWork Student NextWork.org Presentation tier For the presentation tier, we will set...
2. The second half of the code is all about sending responses back e.g. success messages, item not found, error messages. The code uses AWS SDK, which is a set of tools that makes it easier for de...
3. NextWork.org Fetch Data with AWS Lambda Natasha OngNatasha Ong, NextWork Student NextWork.org Introducing Today's Project In this project, we will demonstrate how to use AWS Lambda to ...
4. DynamoDB is a NoSQL database, so it is very fast at retrieval and flexible for data storage. The partition key is user\_id, which means the key identified for each data is its user\_id. In our DynamoDB...

Enter your message here

Run

Test Knowledge Base

Generate responses

Llama 3.1 70B I... v1  
US Meta Llama 3.1 70B Instruct

What is NextWork?

NextWork is an organization that provides projects and resources for students and individuals to work on, with a mission to help everyone find a job they love.<sup>[1]</sup>  
<sup>[2]</sup> [Show details >](#)

Enter your message here

Run

## For More References :

[https://learn.nextwork.org/projects/ai-rag-bedrock?](https://learn.nextwork.org/projects/ai-rag-bedrock?gl=1*hmudkx*_ga*MjAzODY1Mzc3My4xNzI5NTMwMDAy*_ga_P3ZJGC0XCG*MTc0MTM2NDE0Mi4xNDcuMS4xNzQxMzY1OTQyLjAuMC4w&track=high)

[\\_gl=1\\*hmudkx\\*\\_ga\\*MjAzODY1Mzc3My4xNzI5NTMwMDAy\\*\\_ga\\_P3ZJGC0XCG\\*MTc0MTM2NDE0Mi4xNDcuMS4xNzQxMzY1OTQyLjAuMC4w&track=high](https://learn.nextwork.org/projects/ai-rag-bedrock?gl=1*hmudkx*_ga*MjAzODY1Mzc3My4xNzI5NTMwMDAy*_ga_P3ZJGC0XCG*MTc0MTM2NDE0Mi4xNDcuMS4xNzQxMzY1OTQyLjAuMC4w&track=high)