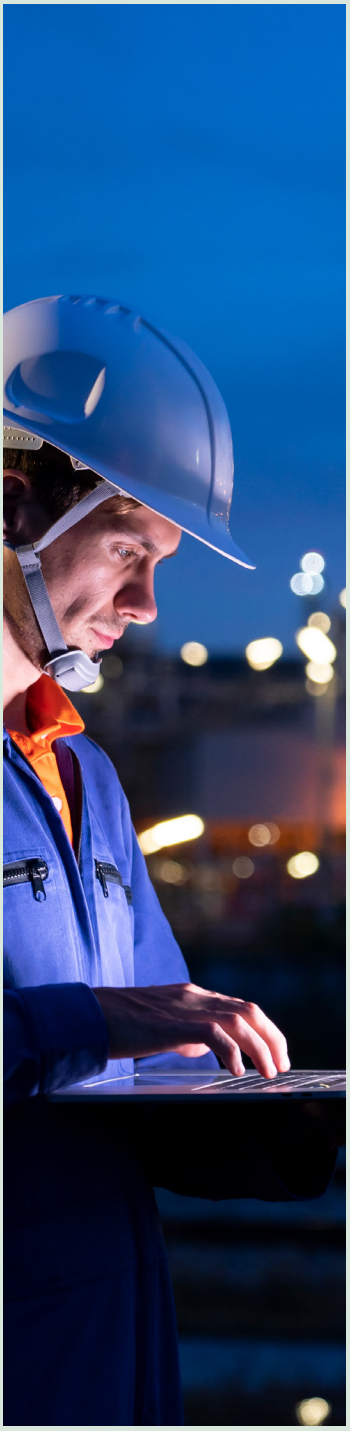
Infosys topaz

WHITE PAPER



INFOSYS TOPAZ™ UNLOCKS INSIGHTS WITH ADVANCED RAG PROCESSING FOR OIL & GAS DRILLING DATA

aws

Infosys®
Navigate your next

Drilling operations in the oil and gas industry generate vast amounts of complex technical data, presenting significant challenges in data processing and knowledge extraction. From detailed well completion reports and drilling logs, to intricate lithology diagrams, these documents contain crucial information that drives operational decisions and strategic planning.

However, traditional document processing methods struggle with the industry's unique challenges: highly technical terminology, complex multi-modal data formats, and interconnected information spread across various document types. This often results in inefficient data extraction, missed insights, and time-consuming manual processing.

Infosys Topaz™, an AI-first offering that accelerates business value for enterprises using generative AI, can integrate AWS generative AI capabilities to help oil & gas companies with rapid prototyping and future-proofing of solutions and ideas.

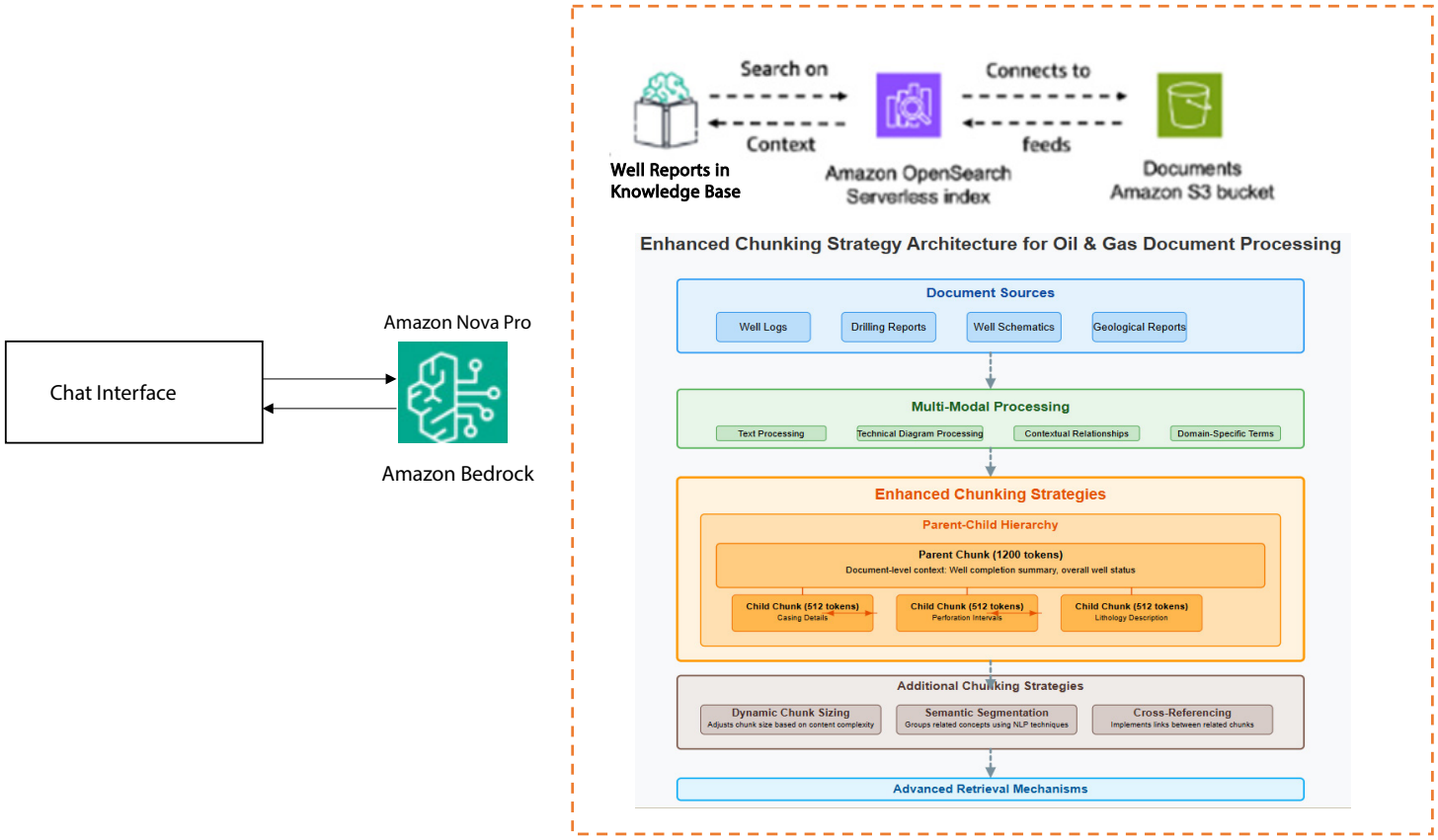
Powered by Infosys Topaz's generative AI capabilities, this solution addresses these challenges head-on through an innovative application of Retrieval-Augmented Generation (RAG) technology, specifically tailored for the oil and gas sector. Unlike conventional RAG systems, the solution addresses the unique challenges of processing domain-specific multi-modal data found in technical documentation such as well completion reports, drilling logs, and lithology diagrams.

The solution can handle multi-modal data sources, seamlessly processing text, diagrams, and numerical data while maintaining context and relationships between different data elements. This specialized approach enables organizations to unlock valuable insights from their technical documentation, streamline their workflows, and make more informed decisions based on comprehensive data analysis.

In this blog post, we provide insights on the solution and walk you through different approaches and architecture patterns explored during the development.

An Overview

The following is a high-level overview of the solution's architecture: -



This solution is built using AWS services, allowing for easy scalability and cost-effectiveness. It uses distributed processing to handle large volumes of data, ensuring that the system can handle a high volume of requests without compromising performance. The real-time indexing system allows for new documents to be incorporated into the system as soon as they are available, ensuring that the information is always up-to-date.

The solution introduces several technical innovations including:

A hierarchical parent-child chunking architecture that preserves document structure and contextual relationships

Multi-vector retrieval mechanisms with separate embedding spaces for textual and visual data

A hybrid search integrating dense and sparse vector approaches

Below are some of the Key components of the solution:

Document Processing:

PyMuPDF for PDF parsing, OpenCV for image processing

Embedding Generation:

Hugging Face Transformers library

Vector Storage:

Amazon OpenSearch Serverless for hybrid search capabilities

Model:

Amazon Nova Pro model for domain-specific response generation

Reranking:

BGE re ranker

Amazon Q Developer:

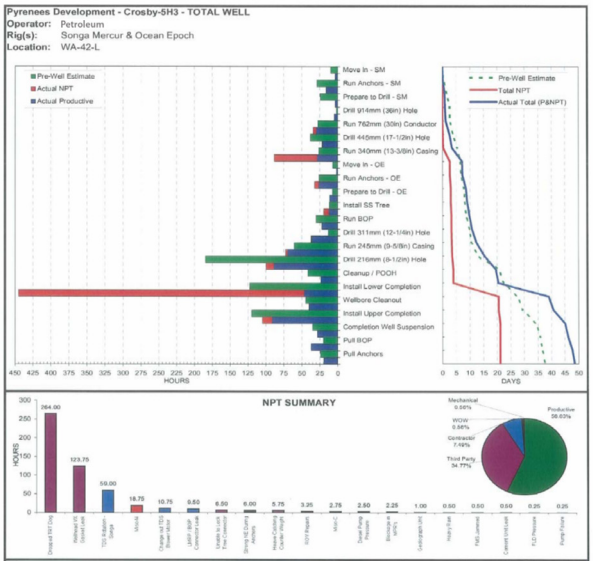
AI powered coding assistant for development

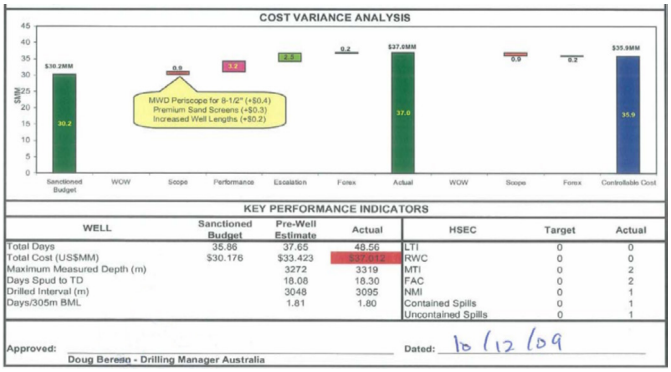
There were many approaches used during the build phase to get the desired accuracy. Let's discuss these approaches in detail.

Knowledge Bases Setup

Below are some sample images from the oil and well drilling reports.

4.1 PERFORMANCE CHART – TOTAL WELL





KEY PERFORMANCE INDICATORS

WELL	Sanctioned Budget	Pre-Well Estimate	Actual	HSEC	Target	Actual
Total Days	35.86	37.65	48.56	LTI	0	0
Total Cost (US$MM)	$30.176	$33.423	$37.012	RWC	0	0
Maximum Measured Depth (m)		3272	3319	MTI	0	2
Days Spud to TD		18.08	18.30	FAC	0	2
Drilled Interval (m)		3048	3095	NMI	0	1
Days/305m BML		1.81	1.80	Contained Spills	0	1
				Uncontained Spills	0	1

Approved: Doug Beresio - Drilling Manager Australia Dated: 10/12/09

7.3.2 8.5IN HOLE SECTION DRILLING DATA

BIT INFORMATION FORM

Date In	08-Aug-09													Date Out	14-Aug-09		
Well Name.	Bit No.		Size (mm)		BHA No.			Make		Type				Serial No.	IADC Code		
CRO-5H3	4		216		4			Reed		RSR616M-B28				222997	M422		
Depth In (mBRT)	Depth Out (m)				Intvl Drilled (m)			Av. ROP (m/hr)		Hrs Drilling				% Slide / Rotary	WOB (t)		
1,470.00	3,319.00				1,849.00			28.8 m/hr		64.3				0 / 100	2.7-6.8t		
m^3/min	Pressure (kPa)				Jet Vel			Jet Impact		TFA (in²)				Nozzles	Tot TBRs (krevs)		
	18,600				66 m/sec			632 lbf		1.035				6 x 15	1013		
Mud Type	Mud Weight				PV / YP			Torque (kNm)		RPM (Surface)				TBR (krevs)	Tot IADC Hrs		
FLOPRO NT	1.16				25 / 40			16.3-27.2		140				1013	82.75		
Dull Bit Grading	I	O	D	L	B	G	O	R	Starting Survey	Depth:	1480.92m	Inc.	87.79°	Az.	43.17°		
	3	4	BT	A	X	I	CT	TD	Last Survey	Depth:	3298.4m	Inc.	88.22°	Az.	45.23°		
BHA #3	REED HYCALOG RSX616M-B28 BIT (SERIAL No. 222997) - XCEED675 - PERSICOPE 675 (RES-APWD-GR) - ILS - TELESCOPE 675 NF (IWOB-MVC-GR-D&I) - ADN-6(DENSITY-POROSITY)-NMDC-HYDRA JAR.																
BHA Length	63.59 m				BHA Weight			15.20 t		Wt below Jar				11.40 t			
Comments:																	

Performance:

Formation	Interval (m)	Interval Hrs	Av Inst. ROP	ROP Range	ROP Inc. Conns
Barrow Group	1470-1696	6.1	37.1 m/hr	40-60	
Muderong	1696-1805	12.6	8.7 m/hr	3-10	
Barrow Group	1805-3319	45.6	33.2 m/hr	25-45	

Bit Condition: Broken teeth appear to have resulted when drilling plug set as the broken teeth had been rounded. Extensive chipping and damage to teeth in inner row. De-lamination present on 2 x cutters on inner row. Gauge was still intact; gauge protector worked well.

Image Sources: Wells Search | NEATS

- 1000+ images were fed into Amazon Nova Pro model, a multi modal language model, and inferenced. Iterative and reflective prompting was used to feed the model with the inference and the image over and again until a detailed description was obtained.
- The text content of the drilling reports along with these image descriptions, preserving the original document structure, were ingested into Amazon's OpenSearch Serverless vector DB.
- Amazon Titan Text Embedding v2 model was used with fixed size chunking of 1500 tokes with 100 tokens overlap for the text content, no chunking for image content, and implemented RAG.

The model worked well with text questions but was less effective with image-related questions and finding information from images. The lack of a chunking strategy for images resulted in the entire description of each image ingested as a single unit into the search index. This made it difficult for the embedding model to pinpoint exact locations of specific information, especially for technical terms that might be buried within longer descriptions.

The section below elaborates the different RAG approaches.

RAG Exploration

The table below outlines alternative strategies followed to overcome previous limitations.

RAG Strategy	Outcome	Limitations & Key Learnings
Multi Vector embeddings with ColBERT - Leverage a vision model to create multi-vector embeddings for each image - Use ColBERT embedding model for fine-grained text representations - Convert user queries into embeddings - Calculate similarity scores between query and document embeddings - Store embeddings using tensor-based storage - No chunking	- Difficulty in storing and managing complex ColBERT embeddings in traditional vector stores - Cumbersome debugging and analysis of retrieved documents - Struggled to select proper document pages even with context-rich queries	- This approach highlighted the potential of advanced embedding techniques for image-based document retrieval. However, it also underscored the challenges in implementing and managing such a system effectively, especially in a domain as complex as oil and gas. Advantages were - - Innovative use of vision models for document understanding - Fine-grained representation of visual and textual content
Fixed Chunking with Amazon Titan Embeddings More traditional text-based approach, we introduced a fixed chunking strategy. - Convert PDF pages to images - Process images to extract text content - Implement fixed chunking strategy (500 tokens per chunk) - Utilize Amazon Bedrock knowledge Bases with OpenSearch vector storage - Upgrade to Titan embedding v2 - Retain Amazon Nova Pro model	- Improved ability to find and retrieve information based on technical keyword searches - More focused chunks allowed for more accurate representation of specific concepts	- Struggled with providing comprehensive, long-form answers - Rigid chunking sometimes split related information across different chunks This approach demonstrated the importance of balancing chunk size with information coherence. It improved our ability to handle technical terms but highlighted the need for more sophisticated chunking strategies to maintain context.
Parent-Child Hierarchy with Cohere Embeddings Building on our learnings, we introduced a more sophisticated chunking strategy using a parent-child hierarchy. - Convert PDF pages to images and extract text - Implement parent-child chunking hierarchy: - Parent chunks: 1500 tokens - Child chunks: 512 tokens - Switch to Cohere English embeddings - Retain Bedrock knowledge base and OpenSearch vector storage - Continue using Amazon Nova Pro model	- Balanced need for context with ability to pinpoint specific information - Significantly improved ability to answer a wide range of queries - Maintained context while offering precise information retrieval	This approach showed that careful structuring of content could significantly enhance the performance of both embedding and QnA models. The parent-child structure proved particularly effective for handling the complex, nested nature of oil and gas documentation.
Hybrid Search with Optimized Chunking Our final approach retained the advanced features of the previous method while introducing a crucial change in the search methodology. - Convert PDF pages to images and extract text - Implement hybrid search system within Bedrock knowledge base - Retain parent-child chunking hierarchy: - Parent chunks: 1200 tokens - Child chunks: 512 tokens - Continue using Cohere English embeddings and Amazon Nova Pro model - Implement BGE reranker to refine search results	- Combined strengths of semantic search and traditional keyword-based search - Addressed limitations of purely embedding-based searches - Improved handling of specific technical terms and exact phrases	This final approach represents a highly evolved RAG system, offering the best of both worlds: the ability to understand context and nuance through embeddings, and the precision of keyword matching for specific technical queries.

Hybrid RAG Approach and Optimization Strategy- deep dive

Let's explore the key components and strategies that make the final approach effective for oil and gas drilling reports.

Multi-Modal Processing Capabilities: The solution is designed to handle the diverse types of information found in oil and gas documents:

Text Processing:

- Handles complex technical jargon and industry-specific terminology
- Processes numerical data such as well logs, production figures, and geological measurements
- Understands context-dependent terms (e.g., "formation" in different drilling contexts)

- **Technical Diagram Processing:**
- Analyzes well schematics, interpreting different components and their relationships
- Processes seismic charts, identifying key geological features
- Handles complex drilling lithology graphs, extracting depth and formation information

Contextual Relationship Maintenance:

- Associates textual descriptions to corresponding diagrams
- Maintains relationships between different sections of a report (e.g., connecting drilling parameters to observed geological features)

Example: When processing a well completion report, the system can:

- Extract key parameters from the text (e.g., total depth, casing sizes)
- Analyze the accompanying well schematic
- Link textual descriptions of formations to their visual representation in lithology charts

Domain-Specific Optimization: Solution is tailored specifically for the oil and gas industry:

Specialized Vocabulary Handling:

- Comprehensive dictionary of industry terms (e.g., "LOT FIT Values", "Mud Weight Interval", "NPT event")
- Understanding of acronyms and abbreviations (e.g., BOP, TVD, MD)

Unit Conversion and Standardization:

- Automatic conversion between different unit systems (e.g., metric to imperial)
- Standardization of depth measurements (e.g., converting all depths to TVD)

Document Type Recognition:

- Specialized processing for different types of reports:
- Well logs: Extract formation tops, hydrocarbon shows
- Drilling reports: Identify key events, equipment used, drilling parameters
- Geological reports: Extract stratigraphic information, reservoir characteristics

Regulatory Compliance:

- Recognition of regulatory terms and standards
- Flagging of potential compliance issues in retrieved information

Example: When processing a query about "fish left in hole at 5000 ft MD", the system understands:

- "Fish" refers to lost equipment, not an actual fish
- "MD" means Measured Depth
- The relevance of this information to drilling operations and potential remediation steps

Enhanced Chunking Strategies: The hybrid hierarchy chunking strategy is crucial for maintaining context while allowing precise information retrieval:

Hybrid Hierarchy:

- Parent chunks (1200 tokens): Maintain document-level context
- Child chunks (512 tokens): Contain detailed technical information
- Bidirectional relationships preserve information flow

Dynamic Chunk Sizing:

- Adapts chunk sizes based on content complexity
- Larger chunks for context-heavy sections (e.g., executive summaries)
- Smaller chunks for dense technical data (e.g., drilling parameters, well log data)
- Uses natural language processing to identify logical break points in the text

Semantic Segmentation:

- Utilizes NLP techniques to identify and group related concepts
- Ensures that related information stays together within chunks
- Particularly useful for maintaining the integrity of technical descriptions

Cross-Referencing:

- Implements a system of cross-references between related chunks
- Allows the retrieval system to quickly navigate between connected pieces of information
- Essential for maintaining context across large documents

Example: When processing a well completion report, the system might:

- Create a large parent chunk for the overall well summary
- Generate smaller child chunks for specific sections like casing details or perforation intervals
- Dynamically adjust chunk size for the lithology description based on its complexity
- Implement cross-references between the casing schedule and the well schematic description

Retrieval Mechanisms: Hybrid search implementation combines multiple techniques for optimal retrieval:

Multi-Vector Retrieval:

- Creates separate embedding spaces for different content types (text, diagrams, numerical data)
- Uses specialized vectors for technical diagrams that capture both visual and textual elements
- Implements cross-modal retrieval to connect information across different content types

Hybrid Search Implementation:

- Dense vector search for semantic similarity
- Utilizes Cohere English embeddings for nuanced understanding of context
- Sparse vector search for technical terminology and exact matches
- Combined ranking system that weights both semantic and keyword relevance

Contextual Query Expansion:

- Automatically expands queries with relevant industry-specific terms
- Includes synonyms and related concepts (e.g., "casing" might expand to include "liner", "tubing")
- Considers the specific context of the query within oil and gas operations

Temporal and Spatial Awareness:

- Incorporates understanding of well locations and operational timelines
- Allows for queries that consider geographical and chronological contexts

Example: For a query like "recent gas shows in Permian Basin wells", the system would:

- Use dense vector search to understand the concept of "gas shows"
- Use sparse vector search to find exact matches for "Permian Basin"
- Expand the query to include related terms like "hydrocarbon indicators"
- Apply temporal filtering to focus on recent reports
- Utilize spatial awareness to limit results to the Permian Basin area

Response Generation and Validation: The solution goes beyond simple retrieval to ensure accurate and relevant responses:

Reflective Prompting:

- Implements a self-questioning mechanism in the language model
- Prompts the model to critically evaluate its own responses
- Checks for consistency with source documents and industry standards

Technical Accuracy Verification:

- Compares generated responses against a database of known facts and relationships in oil and gas
- Flags potential inconsistencies or improbable statements for human review
- Particularly crucial for numerical data and technical specifications

Response Reranking:

- Utilizes an open-source scoring model
- Evaluates responses based on multiple criteria:
- Technical accuracy
- Contextual relevance
- Information completeness

Adherence to industry standards and best practices

Feedback Loop:

- Incorporates user feedback to continuously improve response quality
- Learns from corrections and clarifications provided by domain experts
- Adjusts retrieval and generation strategies based on successful interactions

Example: When generating a response about drilling fluid properties, the system would:

- Retrieve relevant information from multiple sources
- Cross-check numerical values for consistency
- Use reflective prompting to ensure all critical parameters are addressed
- Apply the reranking model to prioritize the most relevant and accurate information
- Present the response along with confidence scores and source citations

Advanced RAG strategies

To further enhance our system's capabilities, we implemented several advanced RAG strategies:

Hypothetical Document Embeddings (HyDE):

- Generates synthetic questions based on document content
- Creates embeddings for these hypothetical questions
- Improves retrieval for complex, multi-part queries
- Particularly effective for handling "what-if" scenarios in drilling operations

Recursive Retrieval:

- Implements multi-hop information gathering
- Allows the system to follow chains of related information across multiple documents
- Essential for answering complex queries that require synthesizing information from various sources

Semantic Routing:

- Intelligently routes queries to appropriate knowledge bases or document subsets
- Optimizes search efficiency by focusing on the most relevant data sources
- Crucial for handling the diverse types of documents in oil and gas operations

Query Transformation:

- Automatically refines and reformulates user queries for optimal retrieval
- Applies industry-specific knowledge to interpret ambiguous terms
- Breaks down complex queries into series of simpler, more targeted searches

Example:

For a complex query like "Compare the production decline rates of horizontal wells in over the last 5 years", the system would:

- Use HyDE to generate relevant sub-questions about decline rates, horizontal wells, and specific formations
- Apply recursive retrieval to gather data from production reports, geological surveys, and economic analyses
- Route different aspects of the query to appropriate knowledge bases (e.g., separate databases for each)
- Transform the query into a series of more specific searches, considering factors like well completion techniques and reservoir characteristics

Result

Average query response time: <2 seconds

Retrieval accuracy (measured against human expert baseline): 92%

User satisfaction rating: 4.7/5 based on feedback from field engineers and geologists

Conclusion

Powered by capabilities from Infosys Topaz and Amazon Bedrock, our journey in developing this advanced RAG solution for the oil and gas industry demonstrates the power of combining AI techniques with domain-specific knowledge. By addressing the unique challenges of technical documentation in this field, we have created a system that not only retrieves information but understands and synthesizes it in a way that adds real value to operations.

This solution opens myriad avenues for advancement, including integration with real-time sensor data for dynamic information retrieval, enhanced visualization capabilities for complex geological and engineering data, and predictive analytics by combining historical retrieval patterns with operational data.

References

- https://www.infosys.com/services/data-ai-topaz.html
- https://aws.amazon.com/blogs/machine-learning/advanced-rag-patterns-on-amazon-sagemaker/
- https://www.pinecone.io/learn/advanced-rag-techniques/
- https://learn.microsoft.com/en-us/azure/developer/ai/advanced-retrieval-augmented-generation

About the Authors

Meenakshi Venkatesan

is a Principal Consultant at Infosys and a part of the AWS partnerships team at Topaz CoE. She helps in designing, developing, and deploying in AWS environments and has interests in exploring the new offerings and services.

Karthikeyan Senthilkumar

is a Senior Systems Engineer at Infosys and a part of the Advanced AI research team at iCETS. He specializes in AWS services with a focus on emerging technologies.

Yash Sharma

is a Digital Specialist Engineer with Infosys and part of Topaz delivery with a passion for emerging generative AI services. He has successfully led and contributed to numerous generative AI projects. He is always eager to expand his knowledge and stay ahead of industry trends, ensuring that he bring the latest insights and techniques to work.

Contributors

Dhiraj Thakur

is a Solutions Architect with Amazon Web Services. He works with AWS customers and partners to provide guidance on enterprise cloud adoption, migration, and strategy. He is passionate about technology and enjoys building and experimenting in the analytics and AI/ML space.

Keerthi Prasad

is a Senior Technology Architect at Infosys and a part of the AWS partnerships team at Topaz CoE. He provides guidance and assistance to customers in building various solutions in the AWS Cloud. He also supports AWS partners and customers in their generative AI adoption journey.

Ganesh S

Enterprise Architect & Data Scientist at Infosys and a part of Topaz delivery. He has a master's degree in computer engineering & machine learning. He has played multiple roles such as architect, program manager and data scientist building scalable enterprise systems, AI/ML and Gen AI applications on Cloud for Oil & Gas, Healthcare and Financial clients.

Suman Debnath

Associate Principal at Infosys and a part of Topaz delivery. He has played multiple roles such as architect, program manager and data scientist building scalable enterprise systems, AI/ML and Gen AI applications on Cloud for Oil & Gas, Healthcare and Financial clients.