LEARNING GEN AI

- BY SANJANA MCS

AWS Bedrock and Embeddings

Companies everywhere are seeking AI expertise to incorporate generative AI into their future tech stacks. Understanding this technology across different cloud platforms is valuable.

When comparing Azure and AWS:

All OpenAI-related services are available in Microsoft Azure.

AWS offers a variety of models, with pricing based primarily on inference costs and LLM usage.

GEN AI PROJECT LIFE CYCLE

1. Define the use case (Problem statement)
2. Choose the right model (foundation model or custom LLM model)
3. Implement prompt engineering, fine-tuning, and RLHF
4. Deploy the solution
5. Integrate the application (optimize and deploy models)

At this stage, we use LLM ops with AWS, Azure, or GCP for inference purposes.

On the user end, we create an API for the specific use case using a service called **API GATEWAY**. Once created, this API triggers an event that connects to a Lambda function, which then calls the foundation models provided by AWS Bedrock.

AWS BEDROCK

Amazon Bedrock is a fully managed service that offers a choice of high-performing foundation models (FMs) from leading AI companies like AI21 Labs, Anthropic, Cohere, Luma (coming soon), Meta, Mistral AI, poolside (coming soon), Stability AI, and Amazon through a single API, along with a broad set of capabilities you need to build generative AI applications with security, privacy, and responsible AI. Using Amazon Bedrock, you can easily experiment with and evaluate top FMs for your use case, privately customize them with your data using techniques such as fine-tuning and Retrieval Augmented Generation (RAG), and build agents that execute tasks using your enterprise systems and data sources.

AWS SAGEMAKER

In this also we can deploy an entire foundation model.

From lamda function we will access the AWS bed rock or the Sage maker.

We cab use open surce model in AWS bedrocks different mdels are provided like nathropic llama1 llaama2.

1. Cloud watch
2. amazon S3 bucket
3. AWS lamda

Foundation Models Explores : Amazon Titan, Anthropic Claude, and Mistral AI

1. Amazon Titan Text Premier

Amazon Titan provides robust foundation models tailored for tasks such as text generation and embeddings. Among these, **Titan Text Premier** stands out for its balanced performance across multiple applications.

- **Model Name**: Titan Text Premier
- **Key Features**:
    - **Maximum Tokens**: 3,072 tokens
    - **Applications**: Summarization, classification, open-ended Q&A, and information extraction.
    - **Inference Parameters**:
        - **Temperature**: Controls randomness in output (range: 0.0 to 1.0).
        - **TopP**: Restricts response diversity by limiting less probable options.
    - **Strength**: A general-purpose model optimized for high accuracy and utility across text-based tasks.

2. Anthropic Claude 3 Opus

Anthropic Claude excels in natural language understanding and text generation tasks, with its **Claude 3 Opus** model being a flagship offering.

- **Model Name**: Claude 3 Opus
- **Key Features**:
    - **Maximum Token Context**: 200,000 tokens
    - **Applications**: Long-form text analysis, multi-step problem-solving, and coding tasks.

    - **Inference Parameters**:
        - **Temperature**: Adjusts randomness in responses.
        - **Max Tokens to Sample**: Defines the token limit for responses.
        - **Stop Sequences**: Allows the model to stop generating text at specified markers.
    - **Strength**: Industry-leading capability for handling complex, large-context tasks with efficiency.

3. Mistral 7B

Mistral AI specializes in open-weight, high-performance models for natural language processing. The **Mistral 7B** model is notable for its efficiency and advanced architecture.

- **Model Name**: Mistral 7B
- **Key Features**:
    - **Parameters**: 7 billion
    - **Architecture**: Includes Grouped Query Attention (GQA) and Sliding Window Attention (SWA) for faster inference and extended sequence handling.
    - **Applications**: Text summarization, multilingual translation, and general-purpose NLP tasks.
    - **Strength**: Open-source with Apache 2.0 licensing, making it highly accessible for custom use cases. Combines speed and efficiency without compromising performance.

Each model has unique strengths tailored to different use cases:

- **Amazon Titan Text Premier**: A versatile model suitable for general-purpose text-based applications.
- **Anthropic Claude 3 Opus**: Designed for long-context and complex text generation tasks.
- **Mistral 7B**: An efficient, open-source model with advanced architecture, ideal for lightweight and fast NLP tasks.

These models empower developers with state-of-the-art capabilities for diverse applications in natural language processing.

Working of AWS bedrock for an Application

1. Understanding the Requirements

- **Objective:** Define the chatbot's purpose (e.g., customer service, employee assistance, or domain-specific use cases).
- **Target Audience:** Identify the end users and the expected conversational tone.
- **Customization Needs:** Determine the need for company-specific data integration and fine-tuning.

2. Selecting the Right Foundation Model

Choose a suitable model based on your use case:

- **Amazon Titan:** General-purpose text embeddings and language understanding.
- **Anthropic Claude:** Safety-focused conversational AI.
- **Cohere:** Multilingual support for semantic understanding.
- **Stability AI:** Image generation if visual outputs are required.

3. Chatbot Workflow

Components:

1. **User Input:** Text or voice input from the user.
2. **Model Interaction:** Query the Bedrock API to generate responses.
3. **Customization:** Integrate company-specific knowledge using fine-tuning or RAG.
4. **Response Delivery:** Provide natural language responses to the user.

4. Implementation Steps

Set Up Bedrock

- Use AWS Management Console or SDKs (e.g., Boto3 for Python) to access Bedrock.

Query the Foundation Model

- Define the user input and choose the desired model.
- Use Bedrock's API to process the input and generate responses.

Example (In Text):

- Initialize Bedrock client.
- Query a model such as **Amazon Titan Text** using your input, e.g., "What is our company's leave policy?"
- The model will return a response based on the provided input.

Customize the Model

- Use fine-tuning to align the model's responses with company-specific terminology.
- Implement RAG to fetch relevant data from enterprise systems:
    - Store data in **Amazon OpenSearch** or **DynamoDB**.
    - Retrieve and combine it with Bedrock's responses.

Deploying and Monitoring

- Use **AWS Lambda** for serverless deployment.
- Integrate with communication platforms like Slack, Teams, or your website.
- Use **Amazon CloudWatch** to track usage and improve performance.

Use Case in Healthcare

Scenario:

A hospital wants a chatbot to assist patients and healthcare staff with:

- Providing health-related advice.
- Scheduling doctor appointments.
- Offering information about medications and treatments.

Workflow:

1. User: "What are the symptoms of diabetes?"
2. Chatbot uses Bedrock to query the **Amazon Titan** model.
3. Retrieves information from medical databases and clinical guidelines.
4. Responds: "Common symptoms of diabetes include increased thirst, frequent urination, and fatigue. Would you like to schedule a consultation?"

Advancements of Amazon Bedrock

- **No Infrastructure Management:** Focus on development, not maintenance.
- **Customizability:** Fine-tune models or use RAG for domain-specific needs.
- **Scalability:** Handle thousands of users seamlessly.

- **Security and Compliance:** Built-in privacy features for sensitive data.

Summary

Amazon Bedrock simplifies the process of building and deploying a customizable chatbot. By leveraging its foundation models, fine-tuning capabilities, and integration with enterprise systems, companies can create efficient, secure, and user-friendly chatbot applications for various domains, including healthcare.