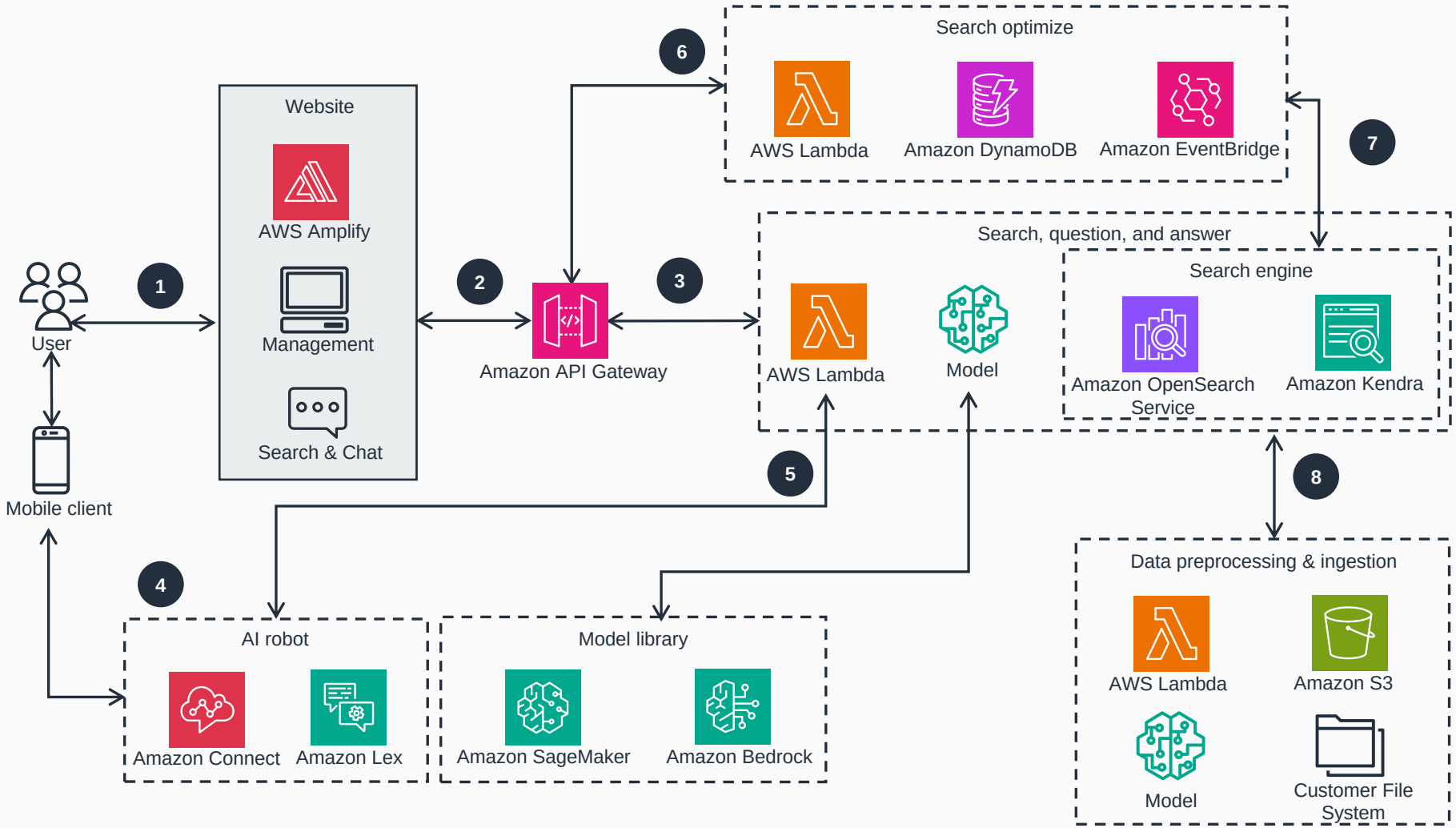
Guidance for Custom Search of an Enterprise Knowledge Base with Amazon OpenSearch Service

This architecture diagram demonstrates how to build an enterprise knowledge base using Amazon OpenSearch Service or Amazon Kendra. These services provide accurate knowledge search and questions and answers based on search method with a machine learning large language model (LLM).



1. The user enters the search query or feedback on the website, which is hosted on **AWS Amplify**.
2. The website passes the input query or feedback to **Amazon API Gateway** and receives the response from **API Gateway**.
3. **API Gateway** passes the input query to the *search, question, and answer* component. This component has **AWS Lambda** integrated with Langchain, an open-source framework. The model library is served by **Amazon Bedrock** or **Amazon SageMaker**. A search engine component can have **Amazon OpenSearch Service** or **Amazon Kendra**. The **Lambda** function will first get the search results from the search engine (either from **OpenSearch Service** or **Amazon Kendra**). The **Lambda** function then inputs the prompt, which combines the query and search results returned from the search engine. It uses the Retrieval-Augmented Generation (RAG) process to optimize the output of the large language model (LLM) and returns the suggested answer from the LLM to **API Gateway**.
4. If the user uses a mobile client, the user can ask a question and get an answer from the artificial intelligence (AI) robot component that has **Amazon Connect** and **Amazon Lex**.
5. **Amazon Connect** transfers the voice into a text query and sends it to **Amazon Lex**. **Amazon Lex** passes the query to the *search, question, and answer* component to get the suggested answer in the same way as Step 3.
6. **API Gateway** passes the feedback to the *search optimize* component, which has **Lambda**, **Amazon DynamoDB**, and **Amazon EventBridge**. The **Lambda** function writes the feedback into **DynamoDB** to help with adjusting the model in the next step.
7. **EventBridge** invokes **Lambda** to train an Extreme Gradient Boosting (XGBoost) model using the feedback stored in **DynamoDB**. Then the model, described by text through decision trees, is deployed to the search engine.
8. The **Lambda** function reads the original file from **Amazon Simple Storage Service** (Amazon S3) or the customer file system. Then it chunks, embeds, and ingests the data into the search engine. The embedding model is hosted on **Amazon Bedrock** or a **SageMaker** endpoint. This serves as the knowledge base for the search engine.

Reviewed for technical accuracy June 25, 2024

AWS Reference Architecture