**Data Ingestion Documentation**

**1. Introduction**

The Data Ingestion component is designed to streamline the process of ingesting PDF documents and converting them into a structured vector database suitable for vector search applications. By leveraging the capabilities of various Google Cloud Platform (GCP) services, the component automates the extraction of text content from PDF files, generates text embeddings using machine learning models, and structures the data into a format compatible with GCP's vector search system. This documentation provides an overview of the component, its objectives, key features, technologies used, usage instructions, and documentation links for further reference.

**2. Project Overview**

The Data Ingestion component addresses the challenge of efficiently indexing and searching through textual data stored in PDF documents. Traditionally, searching through large volumes of unstructured text data can be time-consuming and resource-intensive. By converting PDF documents into a structured vector database format, the project enables fast and accurate similarity search operations, allowing users to quickly retrieve relevant information based on semantic similarity.

**3. Objectives**

The primary objectives of the Data Ingestion project are as follows:

* **Automate PDF to Vector Database Conversion**: Develop automated processes for extracting text content from PDF documents and converting it into a structured vector database format.
* **Generate Text Embeddings**: Utilize machine learning models to generate text embeddings from the extracted text content, capturing semantic information and similarity relationships.
* **Facilitate Efficient Vector Search**: Structure the vector database to enable efficient vector search operations, allowing users to perform similarity search queries and retrieve relevant documents.

**4. Key Features**

Key features of the Data Ingestion project include:

* **PDF to Vector Database Conversion**: The project automates the process of converting PDF documents into a structured vector database format, ensuring compatibility with GCP's vector search system.
* **Text Embeddings Generation**: Leveraging machine learning models provided by GCP's Vertex AI, the project generates text embeddings from the extracted text content, capturing semantic information and similarity relationships.
* **Vector Search Indexing**: The structured vector database is indexed and hosted on GCP's vector search service, enabling efficient similarity search operations for retrieving relevant documents based on semantic similarity.
* **Scalability and Performance**: Built on GCP's scalable infrastructure, the project is designed to handle large volumes of data and perform search operations with high performance and reliability.

**5. Technologies Used**

The Data Ingestion project utilizes the following technologies:

* **Python 3.9**: The primary programming language for development and scripting tasks.
* **Google Cloud Platform (GCP)**: The cloud computing platform used for hosting and managing the project infrastructure.
* **Vertex AI**: GCP's machine learning platform, utilized for generating text embeddings from input text data.
* **GCP Vector Search**: A GCP service for creating and hosting vector search indexes, enabling efficient similarity search operations.
* **Google Kubernetes Engine (GKE)**: GCP's managed Kubernetes service, used as a hosting platform for containerized applications.
* **Docker**: Containerization technology employed for packaging the project components into portable and reproducible containers.

**6. Usage**

To utilize the Data Ingestion project, follow these steps:

1. **Upload PDF Documents**: Upload PDF documents containing the desired text content to the specified folder in the GCP bucket (**gen\_ai\_experiments\_vectordb/data\_source**).
2. **Trigger API**: Hit the designated API endpoint to initiate the conversion process. The project will automatically extract text content from the uploaded PDF documents, generate text embeddings using Vertex AI, and structure the data into a vector database format compatible with GCP Vector Search.

**7. Reference Links**

For additional information and resources, refer to the following documentation links:

* [Vertex AI - Get Text Embeddings](https://cloud.google.com/vertex-ai/docs/generative-ai/embeddings/get-text-embeddings): <https://cloud.google.com/vertex-ai/docs/generative-ai/embeddings/get-text-embeddings>
* [Langchain Documentation - PDF Module](https://python.langchain.com/docs/modules/data_connection/document_loaders/pdf): <https://python.langchain.com/docs/modules/data_connection/document_loaders/pdf>