**KIT-CHAT Content Management Workflow**

**Table of Contents**

* Introduction
* Scope, Audience and Purpose
* Prerequisites
  1. HR Internal Data Collection
  2. Access to PDF and HTML Knowledge Files
  3. Azure Cognitive Search SetUp
  4. Azure OpenAI Integration
  5. Basic knowledge of RAG Pipeline
* Main Solution Components

1. What is Azure Cognitive Search?
2. What is Azure OpenAI?
3. What is the RAG Pipeline?

* Creating Document Indexes

1. Steps to Create a New Index
2. Index Schema Design
3. Data Ingestion from PDF Knowledge Files
4. Document Preprocessing

* Updating Document Indexes

1. Incremental Updates
2. Handling Changes in PDF Knowledge Files
3. Best Practices for Keeping Indexes Updated

* Deleting Document Indexes

1. Reasons for Index Deletion
2. Safe Deletion Practices
3. Impact on Workflows

* KITChat Solution Pipeline

1. Kit-Chat Knowledge Indexing Pipeline
2. Kit-Chat Question Answering Pipeline

**Introduction**

This Proof of Concept (POC) report presents the development and evaluation of a Generative AI-based Question Answering system enhanced with a Retrieval Augmented Generation (RAG) pipeline. Our focus is on the application of this technology to Human Resources (HR) data, showcasing its potential to address HR-related queries and challenges effectively. This report outlines the methodology, content management, key findings, and implications of this innovative data science project on KIT INTERNAL workflows.

**Scope, Audience and Purpose**

The scope of this POC report encompasses the development and evaluation of a Generative AI-based Question Answering system with a Retrieval Augmented Generation (RAG) pipeline tailored for HR data. The target audience for this report includes HR professionals, data scientists, and project stakeholders. The purpose is to demonstrate the feasibility and effectiveness of the system in enabling employees to ask HR-related general and specific queries, thereby providing accurate and timely answers.

**Prerequisites**

Before embarking on this Proof of Concept (POC) report for the Generative AI-based Question Answering system with a Retrieval Augmented Generation (RAG) pipeline on HR data, certain key requirements need to be met. These prerequisites include:

1. **HR Internal Data Collection**: Ensuring access to HR internal data, including policies and work rules, is essential. A comprehensive repository of HR-related content is needed to feed the model effectively.
2. **Access to PDF and HTML Knowledge Files**: Access to HR knowledge files in PDF and HTML formats, stored on SharePoint or similar platforms, is a prerequisite. These files will serve as valuable sources of information for the system.
3. **Azure Cognitive Search Setup**: Setting up Azure Cognitive Search is critical for enabling efficient data retrieval and search capabilities within the HR knowledge repository.
4. **Azure OpenAI Integration**: Integration with Azure OpenAI services is required to leverage generative AI capabilities. This integration enables the system to provide contextually accurate responses to HR queries.
5. **Basic Knowledge of the RAG Pipeline**: Users and stakeholders involved in the project should possess a fundamental understanding of the Retrieval Augmented Generation (RAG) pipeline to facilitate effective collaboration and project comprehension.

Meeting these prerequisites ensures a smooth and productive development and evaluation of the Generative AI-based Question Answering system on HR data with KitChat Application, delivering accurate responses to employee inquiries.

**Main Solution Components**

Azure Cognitive Search

**Definition and Functionality:**

Azure Cognitive Search is a cloud-based search service offered by Microsoft. It empowers organizations to build highly customizable search solutions for their data. In the context of our Generative AI-based Question Answering system with a Retrieval Augmented Generation (RAG) pipeline, Azure Cognitive Search plays a pivotal role in enhancing the search and retrieval process. It's designed to make it easier for users to find relevant information within large datasets quickly. The service offers features like full-text search, filtering, faceted navigation, and more to deliver highly relevant search results.

**Indexing Knowledge Data:**

To enable efficient retrieval of HR-related knowledge in our project, we index the HR data into Azure Cognitive Search. Indexing involves creating a schema or structure that defines how the data should be organized for effective searching. In our case, knowledge data, including policies, work rules, and other HR-related documents, is structured and indexed within Azure Cognitive Search. This indexing process ensures that the information can be rapidly retrieved when HR queries are initiated by employees.

**Retrieval and Answer Generation via RAG Pipeline:**

Our system's Retrieval Augmented Generation (RAG) pipeline is closely connected with Azure Cognitive Search. When an employee submits an HR-related query, the RAG pipeline first leverages Azure Cognitive Search to retrieve relevant knowledge documents from the indexed data. The search results are then used as context for the generative AI model. This model, integrated with Azure OpenAI services, generates accurate and contextually relevant answers to the employee's query.

Azure OpenAI Models

**Definition and Functionality:**

Azure OpenAI Models are a set of cutting-edge, natural language processing (NLP) models developed by OpenAI and integrated into the Azure cloud platform. They are designed to understand and generate human-like text based on the context provided. In the context of our project, the Azure OpenAI Models play a crucial role in enhancing the accuracy and relevance of responses generated by the Generative AI-based Question Answering system with a Retrieval Augmented Generation (RAG) pipeline.

**Retrieving and Utilizing Knowledge Data:**

The integration between Azure OpenAI and Azure Cognitive Search is pivotal in our project. Once Azure Cognitive Search retrieves relevant HR knowledge data in response to an employee's query, this data is used as context by the Azure OpenAI Models. The generative AI model within Azure OpenAI leverages this contextual information to generate accurate and contextually relevant answers.

In this way, the retrieval of HR-related knowledge data from Azure Cognitive Search is seamlessly connected with the capabilities of Azure OpenAI Models. This ensures that the responses provided to employees are not only timely but also contextually appropriate, significantly enhancing the quality of the question-answering system. Employees can thus rely on this system to obtain accurate answers to both general and specific HR-related queries, making it a valuable tool for HR professionals and staff alike.

RAG Pipeline

**Definition and Functionality:**

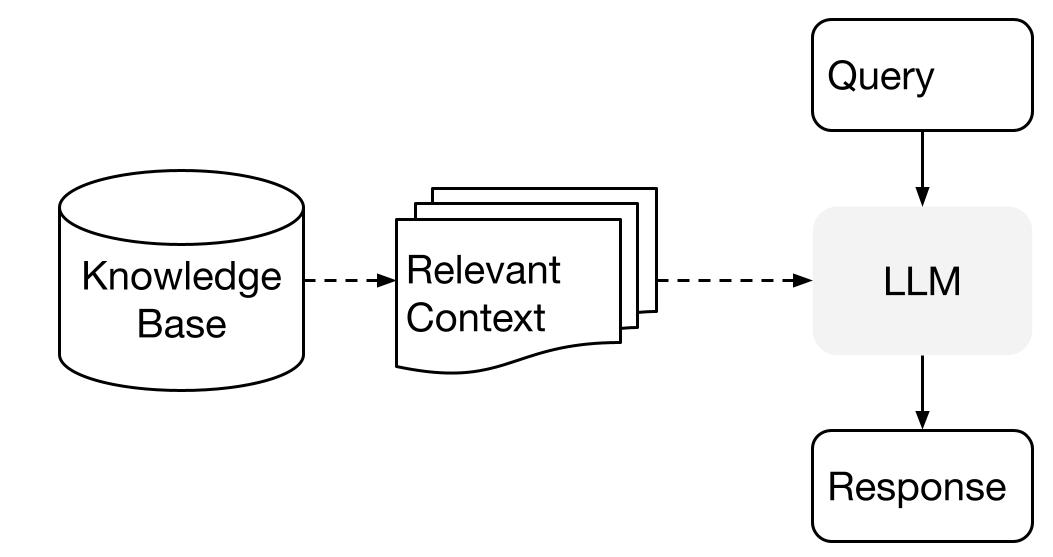
The Retrieval Augmented Generation (RAG) pipeline is a sophisticated approach that combines information retrieval with generative AI. It enhances the accuracy and relevance of responses generated by natural language models. In our project, the RAG pipeline plays a central role in empowering our Generative AI-based Question Answering system to provide accurate and contextually relevant answers to HR-related queries from employees.

**Utilizing RAG on HR Data:**

Our project employs the RAG pipeline to address HR queries effectively. Here's how it works:

**Retrieval Phase**: When an employee submits an HR-related query, the RAG pipeline initiates the retrieval phase. It connects with Azure Cognitive Search, which has indexed HR knowledge data, including policies, work rules, and HR documents. The search component of the pipeline retrieves relevant documents based on the query.

**Generation Phase**: The retrieved documents serve as context for the generative AI model integrated with Azure OpenAI Models. This model generates responses by considering the query and the context from the retrieved documents. This step ensures that the answers provided are both contextually relevant and accurate.



By combining the capabilities of information retrieval with generative AI, the RAG pipeline enriches the quality of responses to HR queries, making our system highly effective in addressing both general and specific HR-related questions. Employees benefit from accurate and timely answers, enhancing the overall efficiency of HR-related processes within the organization.

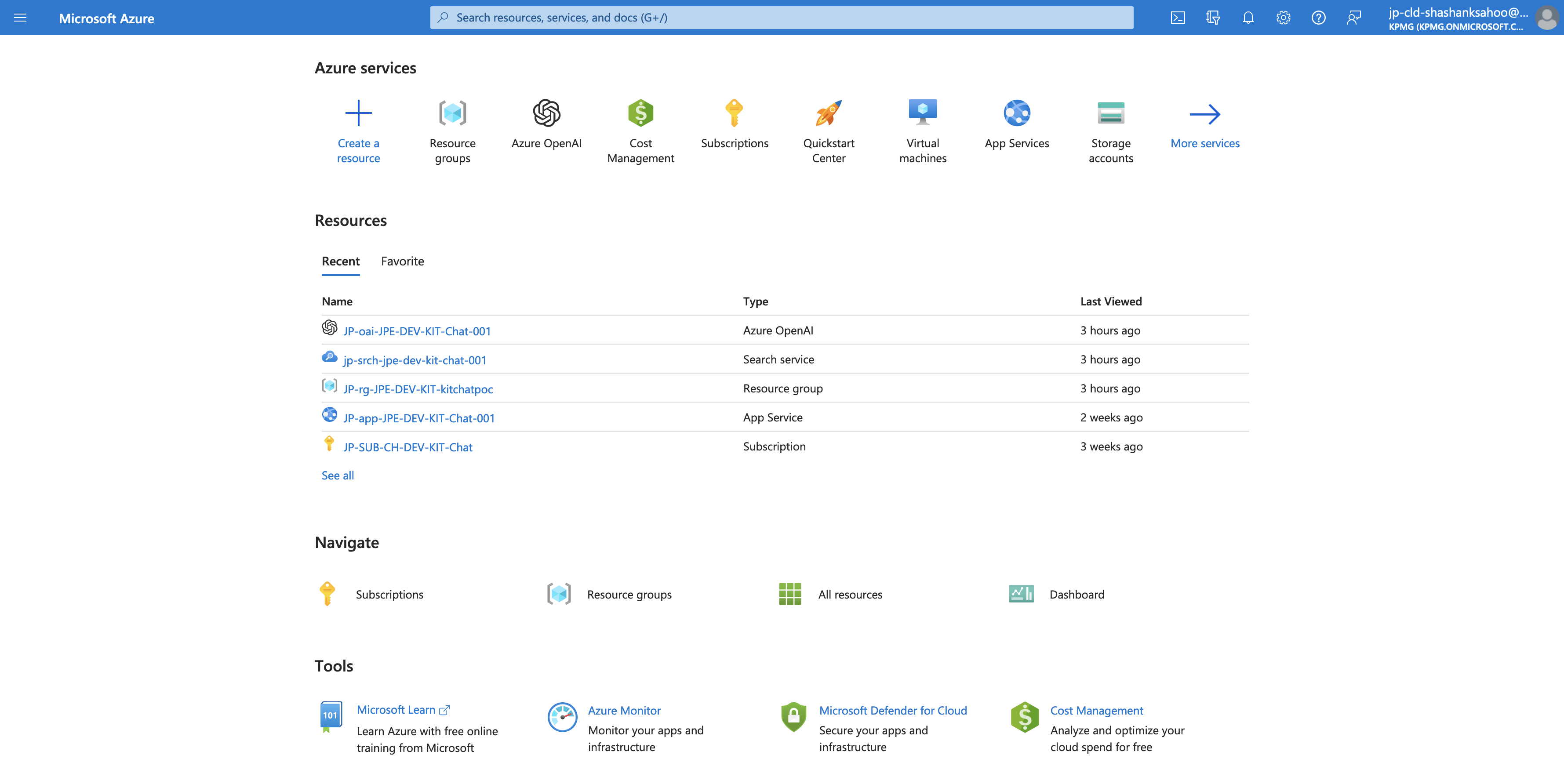
**Creating Document Indexes**

Creating document indexes in Azure Cognitive Search is a fundamental step in enabling full-text search and retrieval of information from your PDF knowledge files. Here are the detailed steps:

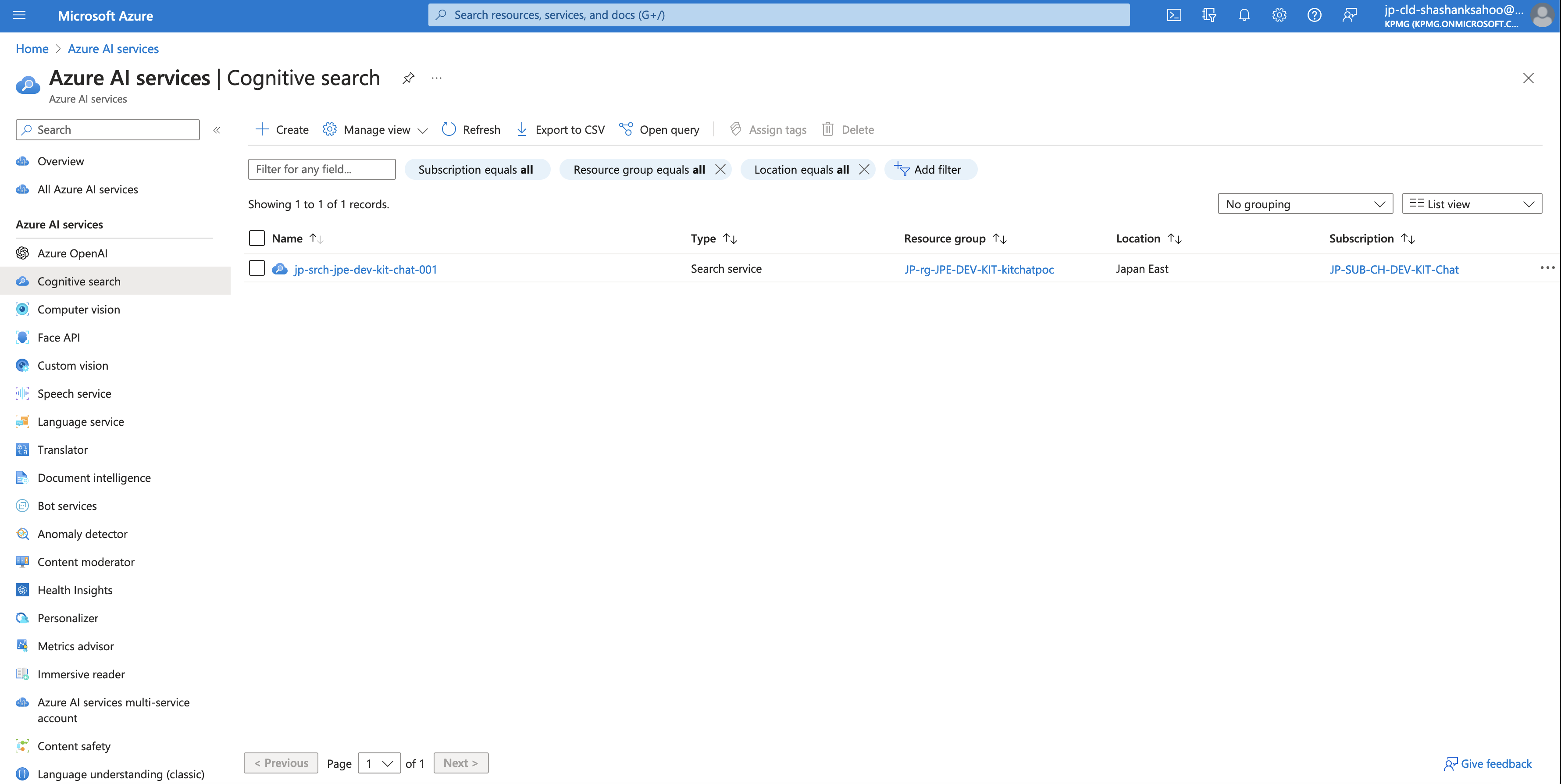
Steps to create a New Index:

1. **Sign into Azure Portal**: Access the Azure portal using your credentials.

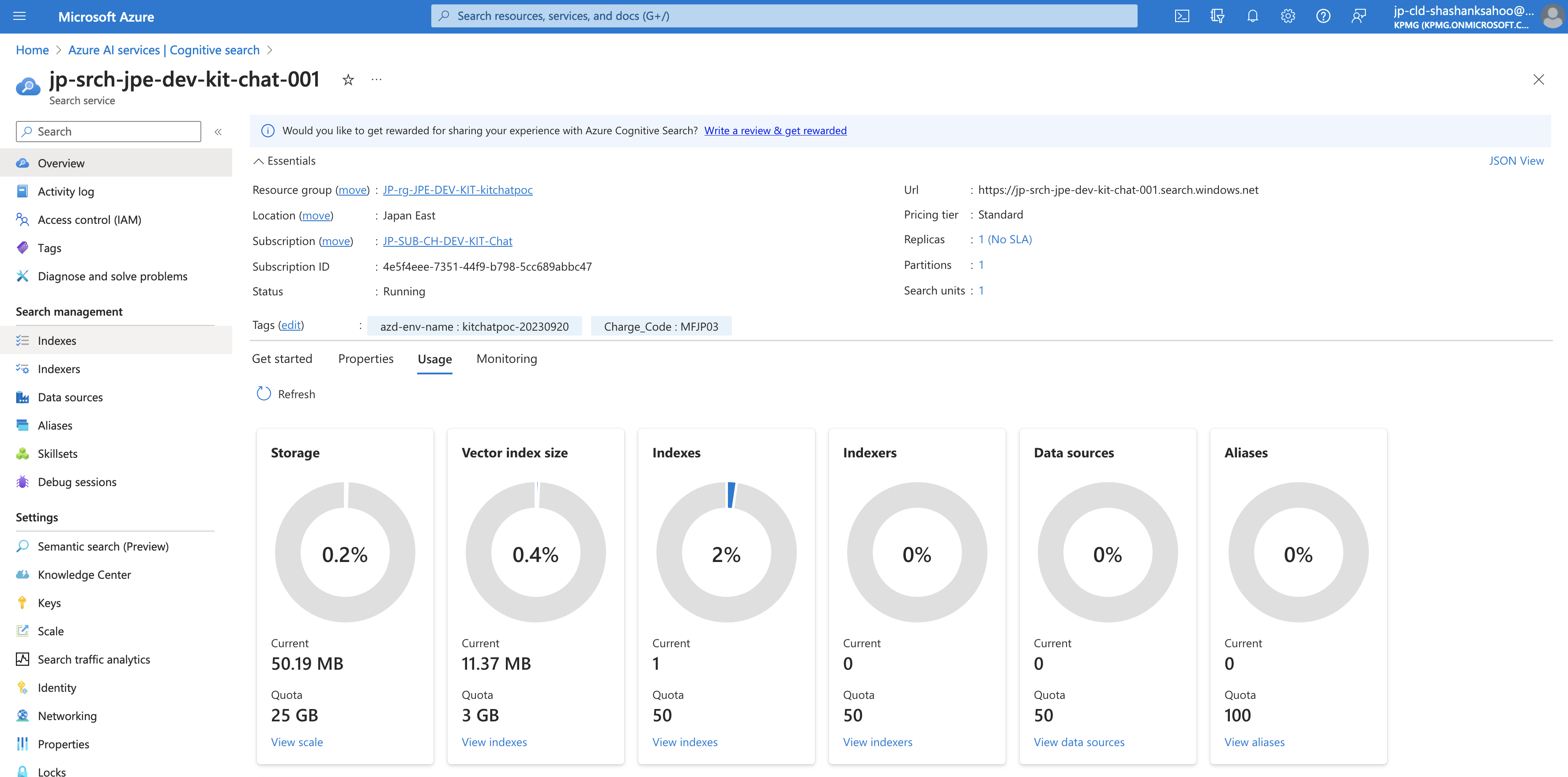
<https://portal.azure.com/#home>



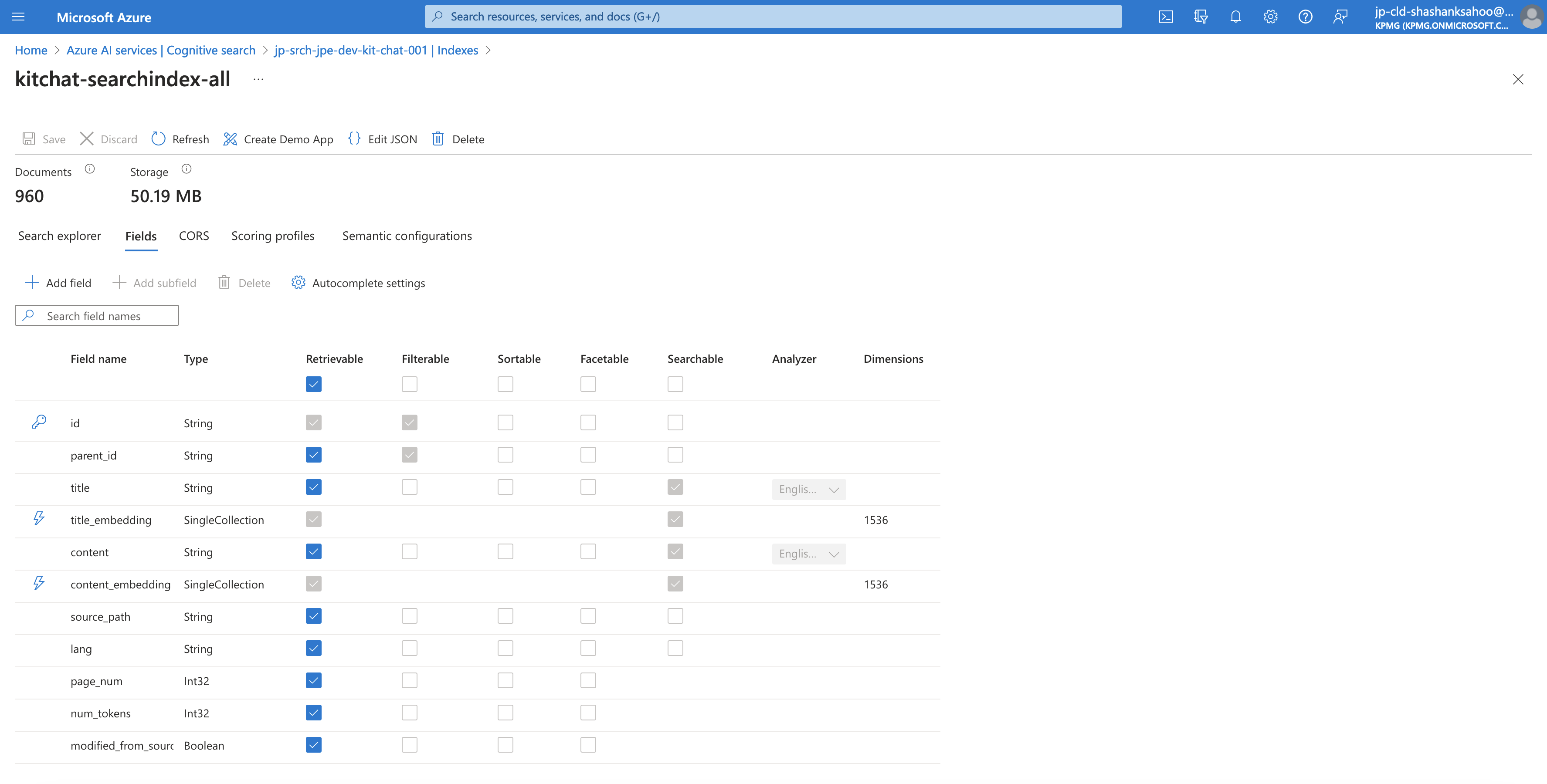
1. **Create a New Search Service**: If you haven't already, create a new Azure Cognitive Search service.



1. **Create an Index**: Within your Search service, create a new index by specifying its name and defining the schema. The schema should include fields that match the attributes of the PDF documents you want to index.



1. **Define Fields**: Define fields for your index, specifying data types (e.g., string, integer, date) and attributes (e.g., searchable, retrievable, filterable) for each field.

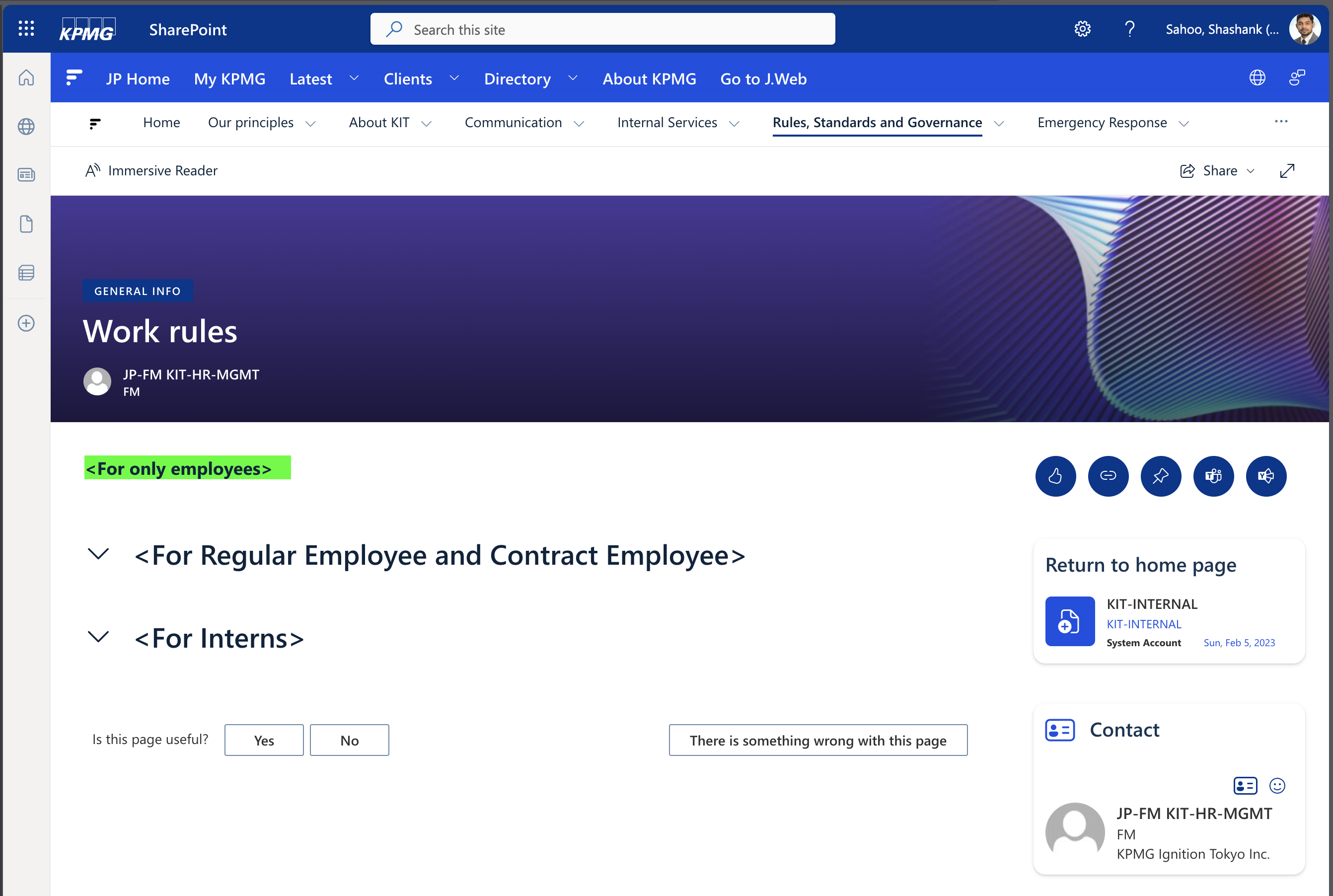


Index Schema Design:

The success of our Generative AI-based Question Answering system with a Retrieval Augmented Generation (RAG) pipeline significantly relies on the precision and efficiency of the knowledge index schema. In designing this schema, we meticulously structured the HR data to facilitate streamlined information retrieval. Specifically, we defined the schema to include key metadata fields such as document title, content, publication date, and category tags. The document title and content are central to the search process, enabling users to find relevant HR documents quickly. The publication date provides a temporal context, which can be crucial for HR policies that change over time. Additionally, category tags help in classifying documents, aiding in specific query refinement. This carefully crafted schema ensures that employees can access the precise HR information they seek, contributing to the system's ability to deliver accurate and tailored answers to HR queries.

Data Ingestion:

Configure data ingestion. You can use Azure Data Factory, Logic Apps, or other methods to ingest data from your PDF knowledge files into the index. But for current *Version 1*, we have ingested the knowledge data manually.

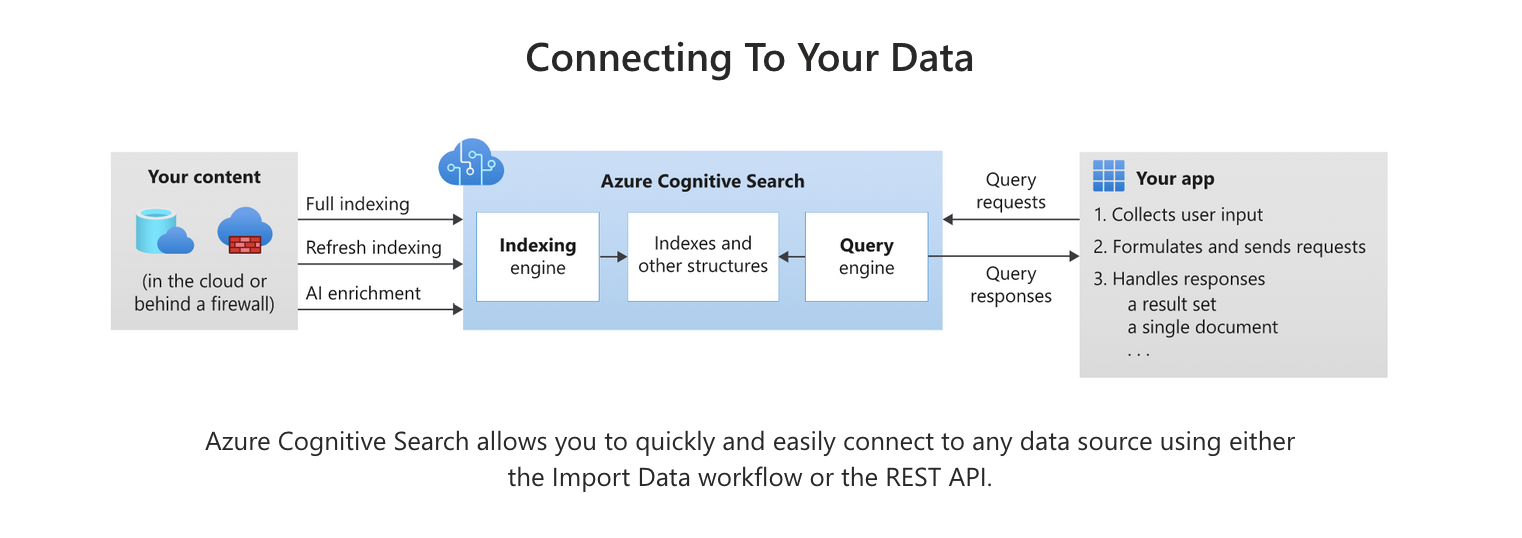


Document Preprocessing:

Implement document preprocessing if needed. This might involve text extraction from PDFs, text cleaning, removing personal information like employee names and language detection. But for current *Version 1*, we have only masked and pseudonymize KIT employees’ names.

**Updating Document Indexes**

Keeping your document indexes up to date is essential to reflect latest changes in your knowledge files. (For current *Version 1*, we are *not supporting* this feature integration)



Incremental Updates:

1. **Change Detection**: Implement a mechanism to detect changes in your PDF knowledge files. This can involve monitoring file modification dates or using event-driven trigger systems.
2. **Data Ingestion**: Use the Azure Cognitive Search data ingestion tools to update the index with the new or modified documents. Azure provides REST APIs for this purpose.
3. **Scheduled Updates**: Consider scheduling regular updates, e.g., daily, or hourly, based on the frequency of changes in your knowledge files.

Handling Changes in PDF Knowledge Files:

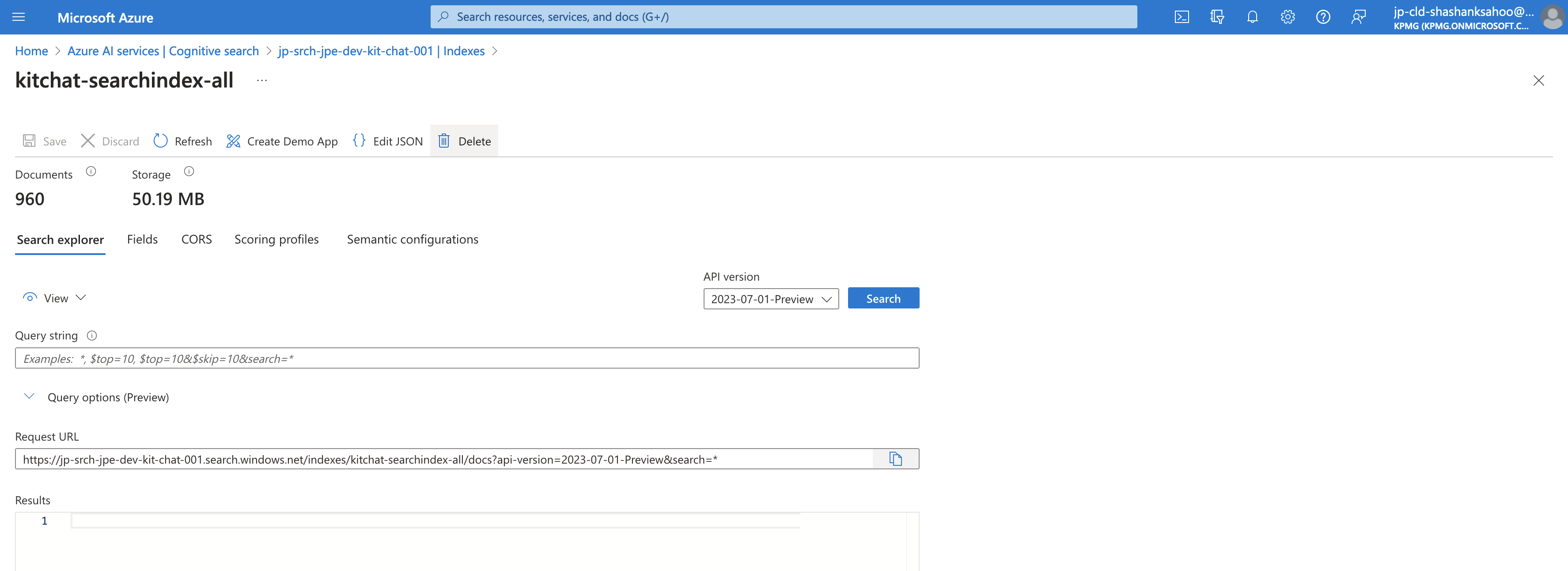
1. **Delta Ingestion**: If your PDF knowledge files are large, consider implementing delta ingestion where only the changes are ingested to save resources.
2. **Conflict Resolution**: Develop strategies for handling conflicts in case the same document is updated simultaneously by different processes.

Best Practices for Keeping Indexes Updated:

In maintaining the knowledge index for our Generative AI-based Question Answering system with a Retrieval Augmented Generation (RAG) pipeline on HR data, certain best practices have been implemented to ensure the system's continued accuracy and relevance. Regular updates of the knowledge index are essential. This involves a systematic approach to add new HR documents, remove outdated information, and reindex the repository periodically. It's crucial to establish version control mechanisms to track changes and maintain historical document versions for reference. Additionally, implementing metadata tagging for newly added documents and periodically revisiting category tags helps in efficient categorization and indexing. Collaborative efforts with HR professionals for content review and updates are essential to keep the knowledge index up to date. These practices collectively ensure that the system continues to provide accurate and timely responses to both general and specific HR queries from employees.

**Deleting Document Indexes**

Deleting document indexes should be done with caution to avoid data loss or disruption of workflows.



Reasons for Index Deletion:

1. **Data Cleanup**: If you no longer need to search certain PDF knowledge files, you can delete their corresponding indexes.
2. **Migration**: When transitioning to a new indexing system or platform.
3. **Data Privacy**: In compliance with data privacy regulations, you might need to delete indexes that contain sensitive information.

Safe Deletion Practices:

1. **Backup**: Before deletion, ensure you have backups of the data if needed for archival purposes.
2. **Permissions**: Restrict access to index deletion to authorized personnel.
3. **Testing**: Test the deletion process in a non-production environment before executing it in the production environment.
4. **Logging**: Keep logs of deletions for auditing purposes.

Impact on Workflows:

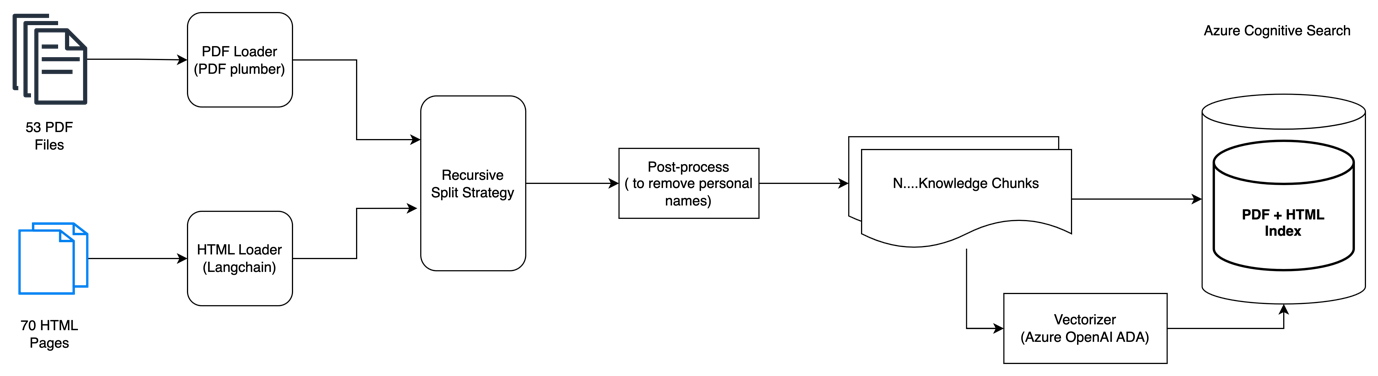
Deleting the knowledge index in our Generative AI-based Question Answering system with a Retrieval Augmented Generation (RAG) pipeline on HR data can have a profound impact on workflow efficiency. It disrupts the system's ability to quickly retrieve HR-related information, leading to delays in providing accurate answers to employee queries. This disruption affects HR professionals' efficiency when responding to employee inquiries, as they will need to manually search for and verify HR documents. Moreover, the absence of an indexed knowledge base can lead to a loss of historical context, hindering the ability to track policy changes and their impacts over time. In summary, deleting the knowledge index significantly hampers the seamless workflow and responsiveness of our HR-related question-answering system, underscoring the importance of diligent knowledge index maintenance.

Remember that Azure Cognitive Search offers a REST API for managing indexes programmatically, so we can integrate index creation, updates, and deletions into your workflows and applications. Always follow best practices and maintain documentation for your specific setup.

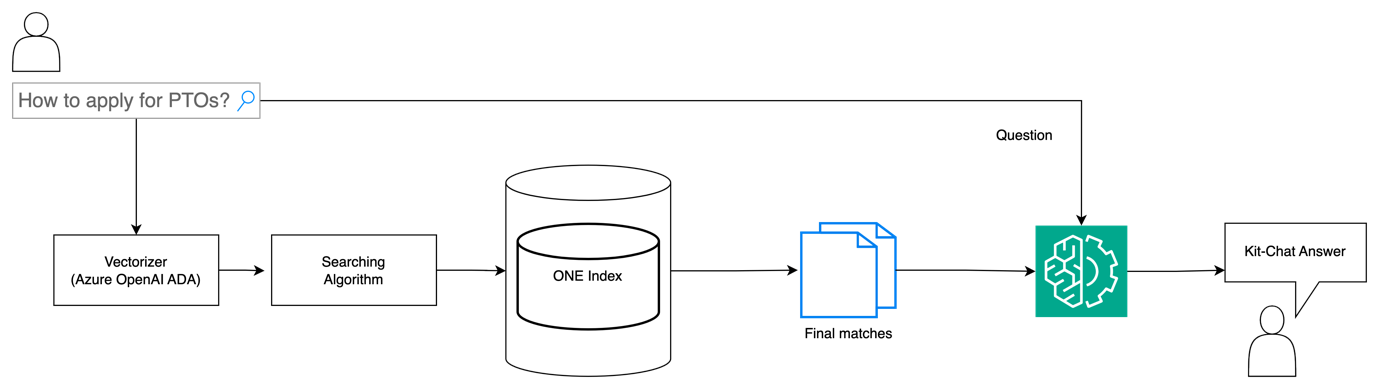
This should give you a detailed starting point for creating, updating, and deleting document indexes in Azure Cognitive Search. Be sure to refer to the official Azure Cognitive Search documentation for the latest guidelines and best practices.

**KITChat Solution Pipeline**

Kit-Chat Knowledge Indexing Pipeline



Kit-Chat Question Answering Pipeline



**Future enhancements**

SharePoint integration

To integrate SharePoint pages with Azure Blob storage for a Kit-Chat system that uses the Azure OpenAI GPT model, one can follow these general steps:

1. **Set Up Azure Resources**:

- Create an Azure Blob storage account to store documents or pages that you want to analyze.

- Set up an Azure Cognitive Search service to enable efficient searching through the content stored in Azure Blob storage.

2. **Upload SharePoint Pages to Azure Blob Storage**:

- Extract and upload the content of SharePoint pages to your Azure Blob storage. Each page could be represented as a document in your storage account.

3. **Define Azure Cognitive Search Index**:

- Define an index in your Azure Cognitive Search service that aligns with the structure of your SharePoint pages. Specify fields such as title, content, URL, etc., based on the information you want to retrieve from the indexed chunks.

4. **Create an Azure Cognitive Search Indexer**:

- Configure an indexer in Azure Cognitive Search to crawl the content stored in Azure Blob storage and populate the search index. Ensure that the indexer is set up to handle the specific format of SharePoint pages.

5. **Use Azure OpenAI GPT Model for Question Answering**:

- Integrate the Azure OpenAI GPT model into your Kit-Chat. You can use the Azure OpenAI API to send queries to the model and receive responses. Formulate your queries based on the content of your indexed documents.

6. **Build an application or API for User Interaction:**

- Develop an application or API that allows users to interact with your question-answering system. This application will send user queries to both Azure Cognitive Search and the OpenAI GPT model.

7. **Implement RAG Pipeline**:

- Implement the RAG (Retrieval-Augmented Generation) pipeline for enhanced question answering. This involves combining information retrieval techniques with the GPT model for more accurate and context-aware responses.

8. **Handle Asynchronous Processes:**

- Account for any asynchronous processes involved in the integration, such as periodic updates of content or retraining of the models. Implement error handling and logging mechanisms to ensure robustness.

9. **Configure Azure Services:**

- Configure Azure services such as Azure Functions, Azure Logic Apps, or Azure Pipelines to automate and orchestrate the integration workflows. This enhances the efficiency and maintainability of the overall system.

10. **Implement Security Measures**:

- Ensure that your application and the data in Azure Blob storage are secured. Implement authentication and authorization mechanisms to control access to sensitive information.

11. **Monitoring and Maintenance**:

- Implement monitoring for both Azure Cognitive Search and the OpenAI GPT model to identify and address any issues promptly. Regularly review and update your search index as new content is added to Azure Blob storage.

12. **Optimization:**

- Optimize your system for performance and cost-effectiveness. Consider caching frequently accessed documents and optimize your queries for efficient searching.

**Note**:

The specifics of the integration can depend on the details of your SharePoint pages, the chosen Azure services, and the requirements of your question-answering system. Always refer to the official documentation for Azure services and OpenAI for the latest and most accurate information.

Citation integration

In the context for a system based on the Retrieval-Augmented Generation (RAG) model using Azure Cognitive Search for information retrieval and Azure OpenAI for response generation, the following steps outline the process to integrate citation footnotes with hyperlinks to answers:

**Identify Relevant Information Sources:**

Identify the sources of information that will be used for citation. This could include documents, databases, or any other repositories configured in Azure Cognitive Search for information retrieval.

**Configure Azure Cognitive Search Index:**

Ensure that the Azure Cognitive Search index is configured to support the retrieval of relevant information. This involves defining fields and settings that align with the nature of the data being searched.

**Retrieve and Process Citations:**

Develop scripts or workflows to retrieve relevant citations from the identified sources using Azure Cognitive Search. Process the retrieved citations to extract necessary information such as title, author, and publication details.

**Generate Hyperlinks for Citations:**

Create hyperlinks for each citation based on the extracted information. These hyperlinks will be embedded in the document and serve as references to the sources of information.

**Incorporate Hyperlinks into Responses:**

During the response generation phase using Azure OpenAI, incorporate the generated hyperlinks into the responses. This step involves modifying the response templates or scripts to include the citation footnotes with clickable hyperlinks.

**Ensure Consistent Formatting:**

Maintain a consistent and user-friendly format for the citation footnotes and hyperlinks across all responses. This ensures a cohesive and professional appearance in the generated content.

**Handle Edge Cases:**

Account for edge cases where a response may not have a direct citation or where the source information is not available. Implement logic to handle such scenarios gracefully.

**Testing and Validation:**

Conduct thorough testing to validate the integration of citation footnotes with hyperlinks. Ensure that the hyperlinks lead to the correct sources and that the formatting meets the desired standards.

**Version Control and Rollback:**

Implement version control for the documentation to track changes related to citation integration. This facilitates easy rollback in case of issues or updates.

Azure App Services Vs Azure Container App services

Azure exposes Azure App service and Azure Container App service to deploy and serve our application. However, the former is Azure managed (**currently being used**) and offer limited capability to debug or fix from app config layer. The debug time thereby increases.

On the other hand, the Container App is more flexible (**self-managed**) and gives more control to debug the issue during deployment. We can consider this if more features are expected to be integrated with App.

Here are some factors that we should consider in future when deciding between Azure App Service and Azure Container Apps:

**Flexibility:** If we want to use any container image to run your application, then Azure Container Apps is a better choice. If we want to use a specific programming language or framework, then Azure App Service is a better choice.

**Control:** If we want more control over the underlying infrastructure, then Azure Container Apps is a better choice. If we want to focus on developing your application and not worry about the underlying infrastructure, then Azure App Service is a better choice.

**Pricing:** Azure Container Apps can be more cost-effective than Azure App Service if you have a large number of applications to host. However, if we have a small number of applications to host, then Azure App Service may be more cost-effective.

**Scalability:** Both Azure App Service and Azure Container Apps provide automatic scaling. However, Azure Container Apps provides more granular control over scaling as we can scale individual containers.