Azure AI Foundry

Workshop 2

03 Avril 2025

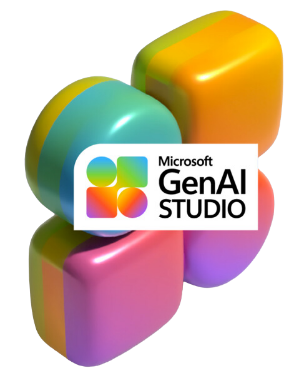


Table des matières

[Introduction to R.A.G. (Retrieval Augmented Generation) 2](#_Toc194571467)

[What is RAG? 2](#_Toc194571468)

[How does RAG work? 2](#_Toc194571469)

[What is an index and why do I need it? 2](#_Toc194571470)

[Lab – Build a RAG solution with the Chat interface 3](#_Toc194571471)

[Prerequisites 3](#_Toc194571472)

[Add your data and try the chat model 4](#_Toc194571473)

[Deploy your web app 10](#_Toc194571474)

[Create a conversational RAG flow 11](#_Toc194571475)

[Evaluating and Deploying LLMs 20](#_Toc194571476)

[Evaluate your Chat flow 20](#_Toc194571477)

[Prepare you chat flow for evaluation 20](#_Toc194571478)

[Create your evaluation flows 21](#_Toc194571479)

[Run the evaluation 24](#_Toc194571480)

[Deploy the RAG flow to an online managed endpoint 29](#_Toc194571481)

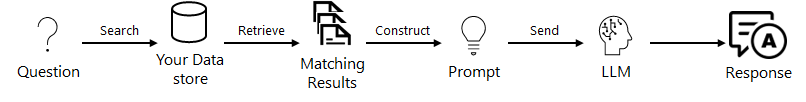
# Introduction to R.A.G. (Retrieval Augmented Generation)

## What is RAG?

Large language models (LLMs) like ChatGPT are trained on public internet data that was available at the point in time when they were trained. They can answer questions related to the data they were trained on. This public data might not be sufficient to meet all your needs. You might want questions answered based on your private data. The public data might simply have gotten out of date. The solution to this problem is Retrieval Augmented Generation (RAG), a pattern used in AI that uses an LLM to generate answers with your own data.

## How does RAG work?

When a user asks a question, the data store is searched based on user input. The user question is then combined with the matching results and sent to the LLM using a prompt (explicit instructions to an AI or machine learning model) to generate the desired answer. This can be illustrated as follows :



## What is an index and why do I need it?

RAG uses your data to generate answers to the user question. For RAG to work well, we need to find a way to search and send your data in an easy and cost-efficient manner to the LLMs. This is achieved by using an index.

An index is a data store that allows you to search data efficiently. This is very useful in RAG. An index can be optimized for LLMs by creating vectors (text data converted to number sequences using an embedding model).

A good index usually has efficient search capabilities like keyword searches, semantic searches, vector searches or a combination of these.

This optimized RAG pattern can be illustrated as follows:

Une image contenant capture d’écran, Police, texte, ligne

Description générée automatiquement

Azure AI provides an index asset to use with RAG pattern. The index asset contains important information like where is your index stored, how to access your index, what are the modes in which your index can be searched, does your index have vectors, what is the embedding model used for vectors etc.

The Azure AI index uses Azure AI Search as the primary and recommended index store. Azure AI Search is an Azure resource that supports information retrieval over your vector and textual data stored in search indexes.

## Lab – Build a RAG solution with the Chat interface

In this lab, you **deploy an enterprise chat web app** that uses your own data with a large language model in AI Studio. Your data source is used to help ground the model with specific data. **Grounding** means that the model uses your data to help it understand the context of your question. You're not changing the deployed model itself. Your data is stored separately and securely in your original data source.

The steps in this lab are:

1. Deploy and test a chat model without your data.

2. Add your data.

3. Test the model with your data.

4. Deploy your web app.

### Prerequisites

* An Azure **subscription**.
* An AI Studio **hub**, **project**, and **deployed Azure OpenAI chat model**. Complete the AI Studio playground [quickstart to create these resources](https://learn.microsoft.com/en-us/azure/ai-studio/quickstarts/get-started-playground) if you haven't already.
* An Azure AI Search service connection to index the sample product data.
* You need a local copy of product data. The Azure-Samples/rag-data-openai python-promptflow repository on GitHub contains sample retail product information that's relevant for this tutorial scenario. Specifically, the product\_info\_11.md file contains product information about the TrailWalker hiking shoes that's relevant for this tutorial example. Download the example Contoso Trek retail product data in a ZIP file to your local machine. [Download the example Contoso Trek retail product data in a ZIP file](https://github.com/Azure-Samples/rag-data-openai-python-promptflow/raw/refs/heads/main/tutorial/data/product-info.zip)
* You need to have Microsoft.Web resource provider registered in the selected subscription, to be able to deploy to a web app. To verify:
  + **Sign in to the Azure Portal**: Go to the Azure Portal and log in with your credentials.
  + **Navigate to Subscriptions**:
    - In the Azure Portal menu, search for **Subscriptions** and select it.
  + **Select Your Subscription**:
    - Choose the subscription where you want to register the resource provider.
  + **Open Resource Providers**:
    - In the left-hand menu, under **Settings**, click on **Resource providers**.
  + **Register Microsoft.Web**:
    - In the list of resource providers, find **Microsoft.Web**.
    - Click on **Register** to register the resource provider.

### Add your data and try the chat model

Follow these steps to add your data in the chat playground to help the assistant answer questions about your products. You're not changing the deployed model itself. Your data is stored separately and securely in your Azure subscription.

1. Go to your project in Azure AI Foundry.
2. Select **Playgrounds** > **Chat** from the left pane.
3. Select your deployed chat model from the **Deployment** dropdown.

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

1. On the left side of the chat playground, select **Add your data** > **+ Add a new data source**.

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

1. Click on “**Upload files**” and “**Upload folder**”:

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

1. Select “**product-info**” folder. The Markdown files are uploaded to the blob storage.

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Une image contenant texte, capture d’écran, nombre, logiciel

Description générée automatiquement

1. Select the Azure AI Search service from the drop-down menu and give your index a name. Leave “Auto select” for the virtual machine.

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

1. Click on “**Add vector search to this search resource**” and select an **Azure OpenAI connection** from the drop-down menu:

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

1. Click on **Create vector index**:

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

1. After a few minutes, your index will be created. It includes vector embeddings.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

1. You can now chat with the model asking the question: "**How much are the TrailWalker hiking shoes?**", and this time it uses information from your data to construct the response. You can expand the **references** button to see the data that was used.

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

### Deploy your web app

Once you're satisfied with the experience in Azure AI Foundry, you can deploy the model as a standalone web application. Publishing creates an Azure App Service in your subscription. It might incur costs depending on the pricing plan you select. When you're done with your app, you can delete it from the Azure portal.

1. Select “**Deploy…as a web app**”:

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

1. Fill in the different cells and click on **Deploy**:

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

1. Wait for the app to be deployed, which might take a few minutes.

### Create a conversational RAG flow

Now you will create a conversational flow using the RAG pattern, start by creating a new flow in the **Prompt Flow** item in the **Tools** section within the **Build** tab.

**Definition**: Prompt flow is a development tool designed to streamline the entire development cycle of AI applications powered by Large Language Models (LLMs). Prompt flow provides a comprehensive solution that simplifies the process of prototyping, experimenting, iterating, and deploying your AI applications. Prompt flow is available independently as an open-source project on [GitHub](https://github.com/microsoft/promptflow), with its own SDK and [VS Code extension](https://marketplace.visualstudio.com/items?itemName=prompt-flow.prompt-flow).

**Benefits of prompt flow:**

With prompt flow in Azure AI Foundry, you can:

* Orchestrate executable flows with LLMs, prompts, and Python tools through a visualized graph.
* Debug, share, and iterate your flows with ease through team collaboration.
* Create prompt variants and compare their performance.

**Enterprise readiness :**

* *Collaboration*: Prompt flow supports team collaboration, allowing multiple users to work together on prompt engineering projects, share knowledge, and maintain version control.
* *All-in-one platform*: Prompt flow streamlines the entire prompt engineering process, from development and evaluation to deployment and monitoring. You can effortlessly deploy their flows as Azure AI endpoints and monitor their performance in real-time, ensuring optimal operation and continuous improvement.
* *Enterprise Readiness Solutions*: Prompt flow applies robust Azure AI enterprise readiness solutions, providing a secure, scalable, and reliable foundation for the development, experimentation, and deployment of flows.

With prompt flow in Azure AI Foundry, you can unleash prompt engineering agility, collaborate effectively, and apply enterprise-grade solutions for successful LLM-based application development and deployment.

**Lab:**

1. Click on **Prompt Flow**:

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

1. Click on **Create** and **Clone** **“Multi-Round Q&A on Your Data”**:

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. Start the automatic runtime by selecting **Start** in the **Runtime** drop-down. The runtime will be useful for you to work with the flow moving forward. The runtime provides the necessary computing resources for the flow to execute. This includes a Docker image with all the required dependencies and packages. The runtime ensures that the flow can run accurately and that any updates to the prompt or code content are properly integrated

Une image contenant texte, capture d’écran, nombre, Parallèle

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**Flow overview**

The first node, **modify\_query\_with\_history**, produces a search query using the user's question and their previous interactions.

Next, in the **lookup node**, the flow uses the vector index to conduct a search within a vector store, which is where the RAG pattern retrieval step takes place.

Following the search process, the **generate\_prompt\_context** node consolidates the results into a string.

This string then serves as input for the **Prompt\_variants** node, which formulates various prompts.

Finally, these prompts are used to generate the user's answer in the **chat\_with\_context** node.

1. Create a search index

Before you can start running your flow, a crucial step is to establish the search index for the Retrieval stage. This search index will be provided by the Azure AI Search service.

In our case, we will create a **Vector index**. To do this, you just need to go back to the project in the **AI Studio**, select the **Indexes** option, and then click on the **New index** button.

Une image contenant texte, capture d’écran, logiciel, ordinateur

Description générée automatiquement

At the **Data source** stage, select the **Upload files/folders** option and upload the **PDF files/surface-pro-4-user-guide-EN.pdf** to the data folder of this lab, as shown in the next screen.

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

In **Index storage**, select the Search Service you created earlier, give a name to your index and select “**Auto select**” for the virtual machine.

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Click on **Create vector index**:

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

It may take about 10 minutes from the time it enters the execution queue until it starts.

Une image contenant texte, capture d’écran, logiciel, diagramme

Description générée automatiquement

Wait until the index status is **Completed** as in the next image, before proceeding with the next steps.

Une image contenant texte, capture d’écran, nombre, logiciel

Description générée automatiquement

1. Let’s configure the **lookup node**. After selecting the **lookup node**, click on **mlindex\_content**.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

A Generate window will appear. In this window, select the **Registered Index** option from the **index\_type** field. Then, choose version 1 of the index you just created. After making these selections, click on **Save**.

Une image contenant texte, reçu, capture d’écran, ligne

Description générée automatiquement

Now, let's go back to the **lookup node**. Select the **Hybrid (vector + keyword)** option from the **query\_type** field.

1. Now you will need to update the Connections of the nodes that link with LLM models. Starting with the Connection in the **modify\_query\_with\_history** node with the **gpt-4o** deployment, as indicated below:

Une image contenant texte, Police, ligne, nombre

Description générée automatiquement

And the Connection for the **chat\_with\_context** node with the **gpt-4o** deployment, as indicated below:

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

1. Everything is now set up for you to initiate your chat flow. Simply click on the blue **Chat** button located at the top right corner of your page to begin interacting with the flow.

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

# Evaluating and Deploying LLMs

In this Lab, you will execute the following steps:

1. Evaluate your Chat flow.
2. Deploy the RAG flow to an online managed endpoint.

## Evaluate your Chat flow

### Prepare you chat flow for evaluation

For the RAG flow that you created earlier to be evaluated, you must include additional information to the output node of this flow, specifically the context used to generate the answer.

This information will be used by the Evaluation Flow.

To do this, just follow these steps:

In the Flows section of **Prompt Flow**, open the **Multi-Round Q&A on Your Data** flow that you created in the previous lab. **This will be the flow we use for evaluation**.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Create a new output named **documents** in the **Outputs** node. This output will represent the documents that were retrieved in the **lookup node** and subsequently formatted in the **generate\_prompt\_context** node.

By creating a new output named **documents** and assigning the formatted documents to it, you're ensuring that the context used to generate answers is explicitly tracked. This context includes the documents retrieved and formatted during the flow.

Assign the output of the **generate\_prompt\_context** node to the **documents** output, as shown in the image below.

Une image contenant texte, ligne, Police, nombre

Description générée automatiquement

Click **Save** before moving to the next section.

### Create your evaluation flows

Still in the **Prompt flow** item in the **Tools** section, click on the blue **Create** button.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Select the **Evaluation Flow** filter and click on **Clone** on the **QnA Groundedness Evaluation** card.

Une image contenant texte, capture d’écran, Police, reçu

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, algèbre

Description générée automatiquement



A flow will be created with the following structure:

Une image contenant texte, diagramme, capture d’écran, Police

Description générée automatiquement

**Definition**: The **Groundedness Evaluation** is a metric used to assess whether the responses generated by a generative AI model are based on the provided source materials. This is particularly important in Retrieval-Augmented Generation (RAG) scenarios, where the model retrieves and uses specific documents or data to generate answers.

**Key Points:**

* **Purpose**: To ensure that the AI's responses are accurate and based on the relevant data sources, rather than being fabricated or unrelated.
* **Process**: During evaluation, the model's responses are checked against the source documents to verify that the information provided is grounded in those documents.
* **Importance**: This helps in maintaining the reliability and trustworthiness of the AI model, ensuring that users receive accurate and contextually relevant information.

Update the **Connection** field to point to a **gpt-4o** deployment in **groundedness\_score** node also update **max\_tokens to 1000** as shown in the next figure.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

After updating the connection information, click on **Save** in the evaluation flow and navigate to the Flows section in **Prompt Flow** item.

Now, you will repeat the same steps described so far to create **two** additional evaluation flows, one **QnA Relevance Evaluation** and another **QnA GPT Similarity Evaluation**.

Une image contenant texte, capture d’écran, reçu, Police

Description générée automatiquement

**Definition:** The **relevance score** is a metric used to measure how well the AI model's responses match the user's query or the intended context. Here's a simplified explanation:

**Key Points:**

* **Purpose**: To evaluate the quality and appropriateness of the AI-generated responses.
* **Calculation**: The score is computed based on how closely the response aligns with the user's query and the provided context. *The higher the score, the more relevant the response is considered to be.*
* **Usage**: This metric helps in assessing and improving the performance of generative AI models by ensuring that the responses are not only accurate but also contextually appropriate

In essence, the **relevance score** helps you understand how effectively your AI model is meeting user needs by providing pertinent and useful answers.

Update the **Connection** field to point to a **gpt-4o** deployment in **relevance\_score** node also update **max\_tokens to 1000** as shown in the next figure.

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

**Definition**: The **similarity score** is a metric used to measure how closely the AI model's generated responses match the expected or reference responses. Here's a simplified explanation:

**Key Points:**

* **Purpose**: To evaluate the degree of similarity between the AI-generated text and the reference text.
* **Calculation**: This score is typically calculated using various methods, such as cosine similarity, which compares the semantic meaning of the texts by converting them into vector representations
* **Usage**: It helps in assessing how well the AI model is performing in terms of generating responses that are semantically similar to the expected answers. This is particularly useful for tasks like text generation, translation, and summarization

In essence, the **similarity score** helps ensure that the AI model's outputs are not only relevant but also closely aligned with the intended responses, improving the overall quality and reliability of the model.

Update the **Connection** field to point to a **gpt-4o** deployment in **similarity\_score** node also update **max\_tokens to 1000** as shown in the next figure.

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

### Run the evaluation

In the Flows section of **Prompt Flow**, open the **Multi-Round Q&A on Your Data** flow that you created in the previous lab. This will be the flow we use for evaluation.

Start the automatic runtime by selecting **Start** in the **Runtime** drop-down.

Select the **Custom evaluation** option in the Evaluate menu.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

**Prompt variants** refer to different versions of a prompt or tool node that have distinct settings.

In the **Prompt\_variants** option, select the option to run only **two variants** to avoid reaching your GPT-4o model quota limit, as shown in the example image below.

Une image contenant texte, capture d’écran, ligne, nombre

Description générée automatiquement

Select **Add new data**.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Upload the file **data.csv**.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

After clicking on **Add** proceed to map the input fields as shown below:

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Select the three evaluation flows you just created.

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

Click on **Next** to set up the **question, context, ground\_truth** and **answer fields** for each evaluation flow. You can see how to do this in the three images below.

Une image contenant texte, Police, ligne, nombre

Description générée automatiquement

Une image contenant texte, nombre, ligne, Police

Description générée automatiquement

Une image contenant texte, logiciel, nombre, capture d’écran

Description générée automatiquement

Click on **Submit** to start the evaluation.

The evaluation process has started. To view all evaluations (one per variant), please navigate to the **Evaluation** section under the **Build** tab.

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

**Groundedness**

* **Use it when:** You're worried your application generates information that isn't included as part of your generative AI's trained knowledge (also known as unverifiable information).|
* **How to read it:** If the model's answers are highly grounded, it indicates that the facts covered in the AI system's responses are verifiable by the input source or internal database. Conversely, low groundedness scores suggest that the facts mentioned in the AI system's responses may not be adequately supported or verifiable by the input source or internal database. In such cases, the model's generated answers could be based solely on its pretrained knowledge, which may not align with the specific context or domain of the given input
* **Scale:**
  + 1 = "ungrounded": suggests that responses aren't verifiable by the input source or internal database.
  + 5 = "perfect groundedness" suggests that the facts covered in the AI system's responses are verifiable by the input source or internal database.

**Relevance**

* **Use it when**: You would like to achieve high relevance for your application's answers to enhance the user experience and utility of your generative AI systems.
* **How to read it**: Answers are scored in their ability to capture the key points of the question from the context in the ground truth source. If the model's answers are highly relevant, it indicates that the AI system comprehends the input and can produce coherent and contextually appropriate outputs. Conversely, low relevance scores suggest that the generated responses might be off-topic, lack context, or fail to address the user's intended queries adequately.
* **Scale**:
  + 1 = "irrelevant" suggests that the generated responses might be off-topic, lack context, or fail to address the user's intended queries adequately.
  + 5 = "perfect relevance" suggests contextually appropriate outputs.

**Similarity**

* Use it when: You would like to objectively evaluate the performance of an AI model (for text generation tasks where you have access to ground truth desired responses). Ada similarity allows you to compare the generated text against the desired content.
* How to read it: Answers are scored for equivalencies to the ground-truth answer by capturing the same information and meaning as the ground-truth answer for the given question. A high Ada similarity score suggests that the model's prediction is contextually similar to the ground truth, indicating accurate and relevant results. Conversely, a low Ada similarity score implies a mismatch or divergence between the prediction and the actual ground truth, potentially signaling inaccuracies or deficiencies in the model's performance.
* **Scale**:
  + 1 = "nonequivalence" suggests a mismatch or divergence between the prediction and the actual ground truth, potentially signaling inaccuracies or deficiencies in the model's performance.
  + 5 = "perfect equivalence" suggests that the model's prediction is contextually similar to the ground truth, indicating accurate and relevant results.

**Note:** you can use the SDK to evaluate your models => [Evaluate with the Azure AI Evaluation SDK - Azure AI Foundry | Microsoft Learn](https://learn.microsoft.com/en-us/azure/ai-studio/how-to/develop/evaluate-sdk)

### Deploy the RAG flow to an online managed endpoint

Open the **Multi-Round Q&A on Your Data** flow that you created in the previous lab.

Now that you have built a flow and tested it properly, it's time to create your online endpoint for real-time inference.

Follow the steps below to deploy a prompt flow as an online endpoint in Azure AI Foundry.

1. Have a prompt flow ready for deployment.
2. Select **Deploy** on the flow editor.

Une image contenant texte, capture d’écran, ligne, Police

Description générée automatiquement

Une image contenant texte, capture d’écran, nombre, diagramme

Description générée automatiquement

Une image contenant texte, Police, nombre, capture d’écran

Description générée automatiquement

You can test it:

Une image contenant texte, Police, capture d’écran

Description générée automatiquement