Azure AI Foundry

Lab

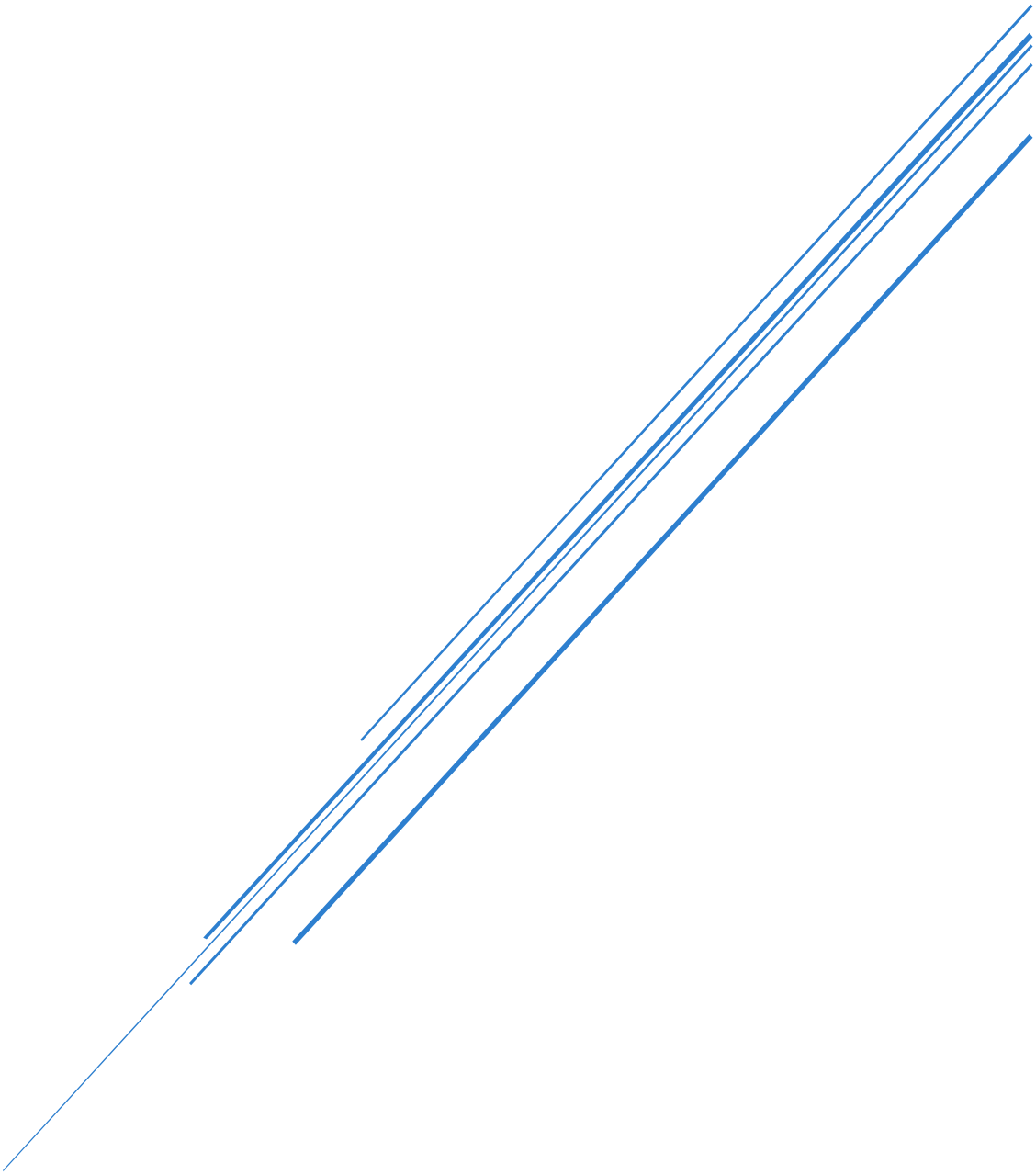


Table des matières

Environment setup

To deploy this lab, an **Azure subscription is required**, where you can create an AI Project along with its AI Hub Resource, a Content Safety service, and an AI Search service.

Lab Steps

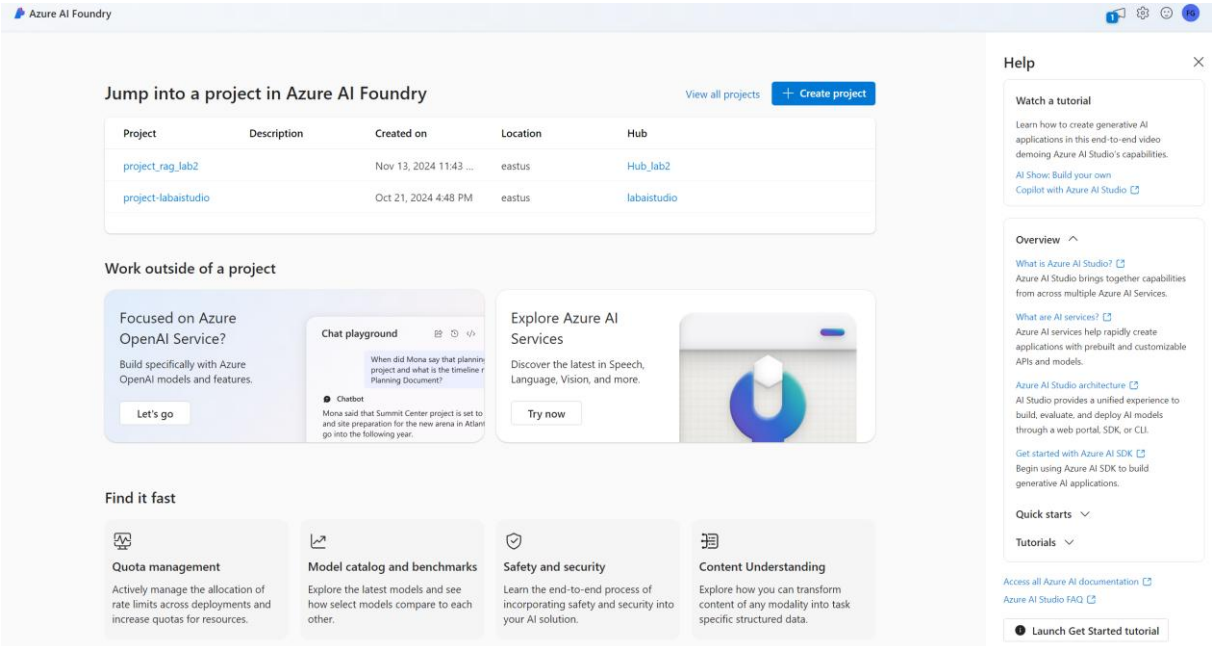
1. Use Azure AI Foundry Playground.
2. Work with an Open Source LLM Model.
3. RAG – connect your data.
4. Create a Prompt Flow flow
5. Evaluate your model
6. Test the prompt in Content Safety.

Create an AI Project and AI Hub Resources

Let's start by creating a project in Azure AI Foundry.

Go to your browser and type: https://ai.azure.com.

Logging with your Microsoft account.



Create a new Project

Definition

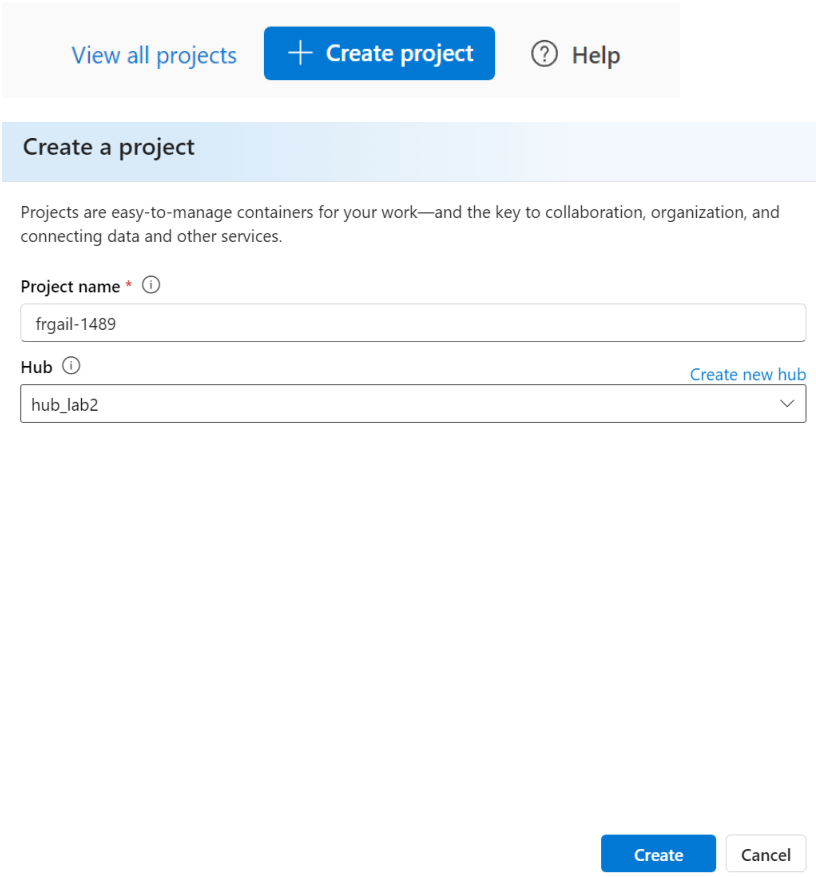
A **project** is an organizational container that has tools for AI customization and orchestration. It lets you organize your work, save state across different tools like prompt

flow, and collaborate with others. For example, you can share uploaded files and connections to data sources.

Multiple projects can use a hub, and multiple users can use a project. A project also helps you keep track of billing and manage access and provides data isolation. Every project uses dedicated storage containers to let you upload files and share it with only other project members when using the 'data' experiences.

Action

- **Click on**: https://ai.azure.com/
- **Click on:** "+ Create project"



- Type a name for your project.
- Click on: "**Create new hub**"

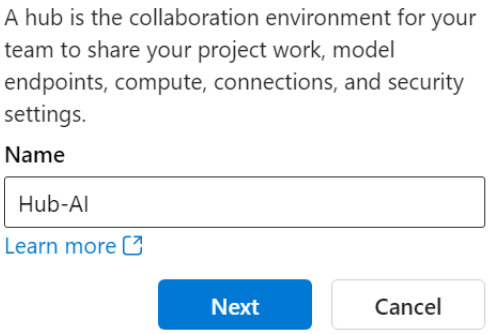
Definition: Hubs are the primary top-level Azure resource for AI Studio and provide a central way for a team to govern security, connectivity, and computing resources across playgrounds and projects. Once a hub is created, developers can create projects from it and access shared company resources without needing an IT administrator's repeated help.

You can create and manage a hub from the Azure portal or from the AI Foundry.

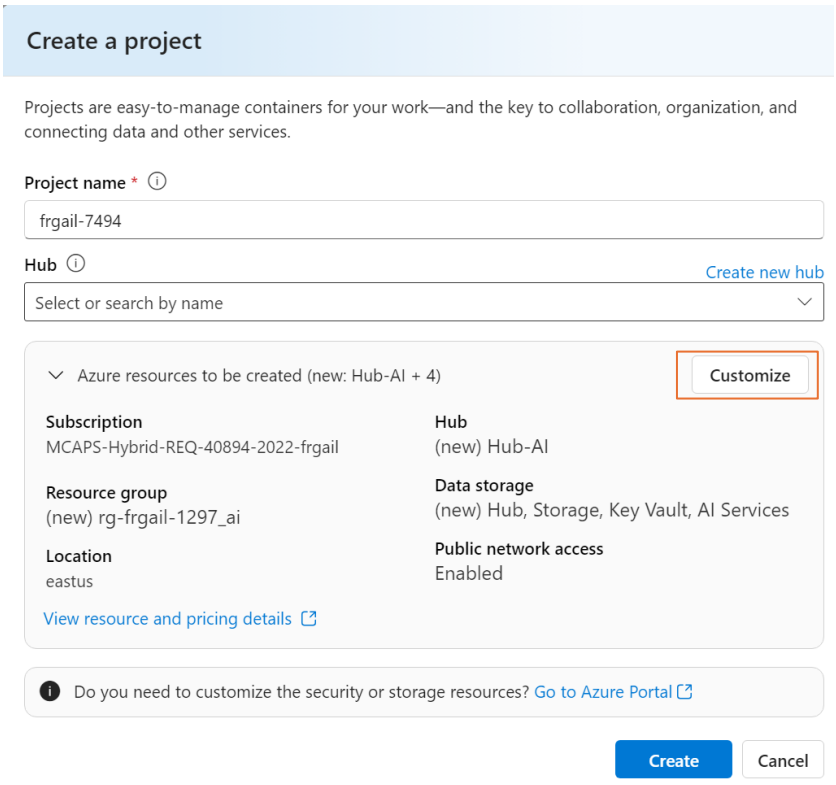
If you want to **create a secure Hub**, you must create the Hub from the portal:

[How to create and manage an Azure AI Foundry hub - Azure AI Foundry | Microsoft Learn]

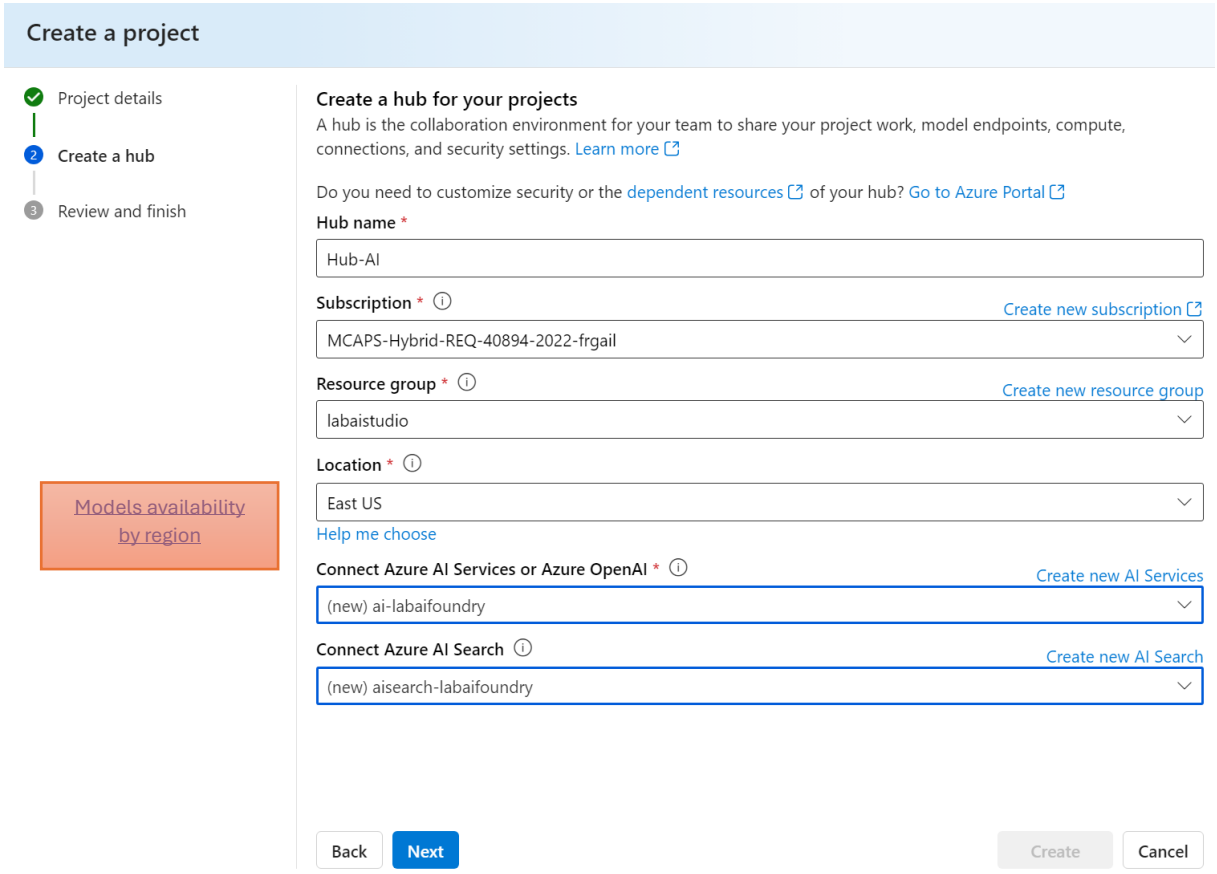
- Type a name for your hub



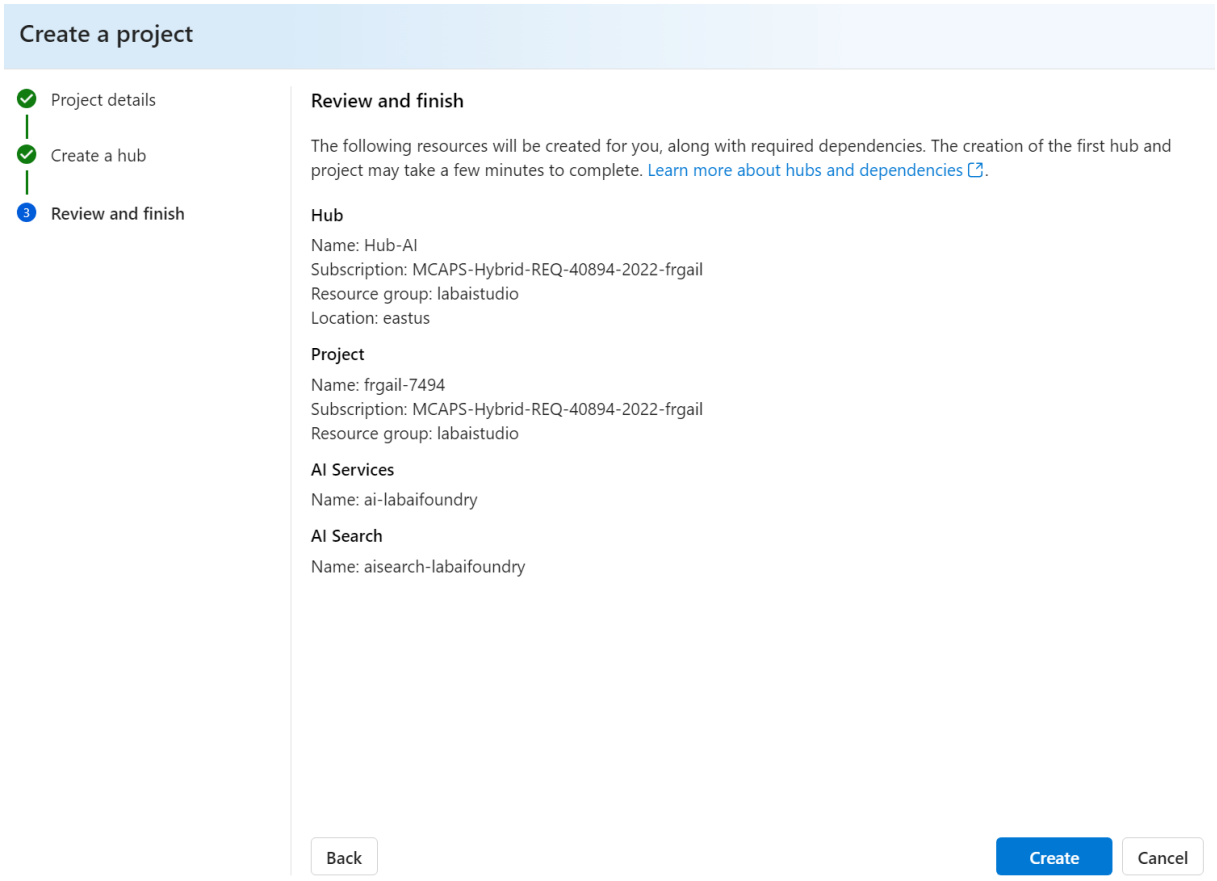
- Click on **Customize**



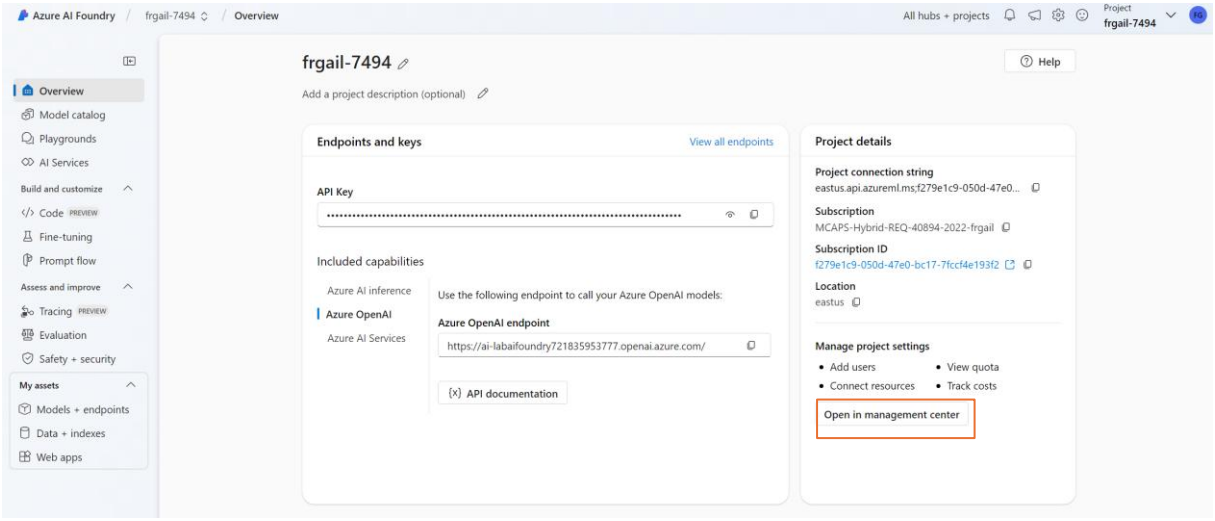
- Fill the cells



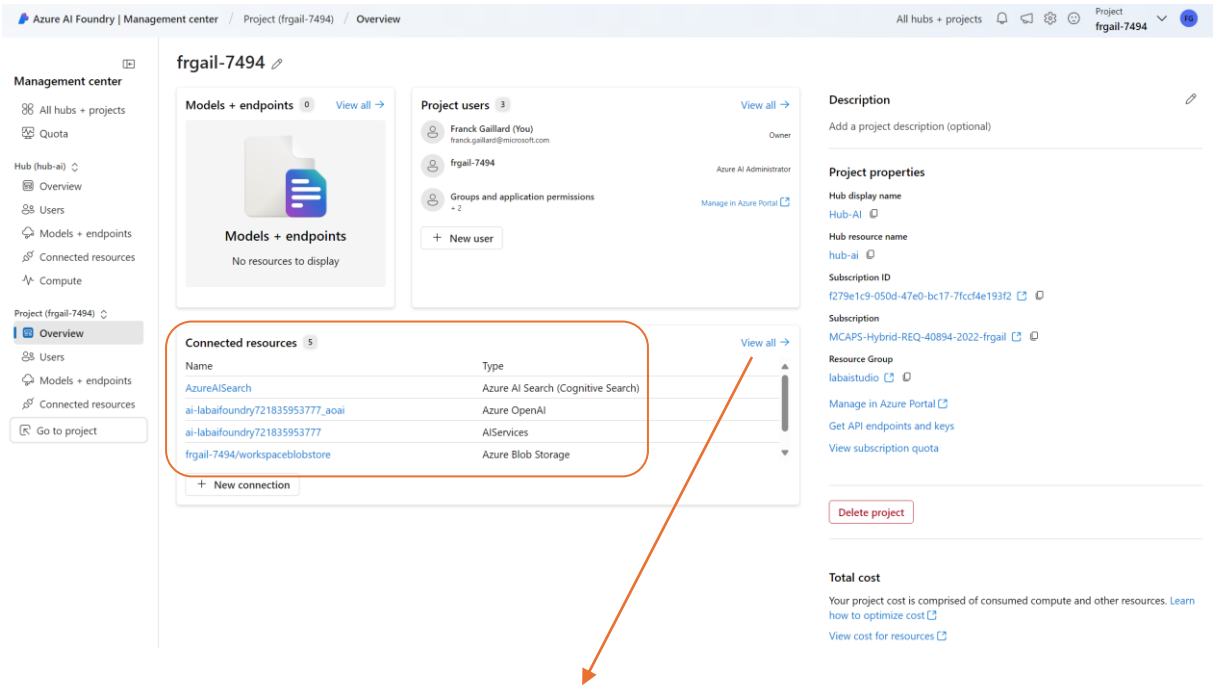
- Click on **Create**



- Click on **Open in management center**



Definition: The **management center** is a centralized control panel within the Azure AI Foundry portal. It streamlines governance and management activities, providing a simplified and centralized experience for cross-functional teams. From the management center, you can manage various aspects such as projects, resources, quotas, usage metrics, access rights, and permissions.



Name	Type	Target	Key	Authentication type
AzureAISearch	Azure AI Search (Cogniti...	https://aisearch-labaifoundry721835953777.se...	.....	API key
ai-labaifoundry721835953777_aoai	Azure OpenAI	https://ai-labaifoundry721835953777.openai.a...	.....	API key
ai-labaifoundry721835953777	AIServices	https://ai-labaifoundry721835953777.cognitive...	.....	API key
frgail-7494/workspaceblobstore	Azure Blob Storage	https://sthubai721835953777.blob.core.windo...	--	SAS
frgail-7494/workspaceartifactstore	Azure Blob Storage	https://sthubai721835953777.blob.core.windo...	--	SAS

4 connected resources have been created:

- **Azure AI Search** to build an index and store vectors.
- **AI Services** to access AI Services (Speech, Language+Translator, Vision, Content Safety)
- **Azure OpenAI service** to access Azure OpenAI models
- **Azure Blob Storage** to store data and artifacts.
    - **Workspace Artifact Store:**
        - Primarily used to store various artifacts related to your AI projects, such as datasets, models, logs, and other files.
        - Each project has its own dedicated storage containers within the workspace artifact store, maintaining data isolation and security.
    - **Workspace Blob Store :**
        - Acts as the default blob storage for the workspace, used for general data storage needs.
        - Typically used for storing large amounts of unstructured data, such as text, images, and binary data.

In summary, the Hub's key points:

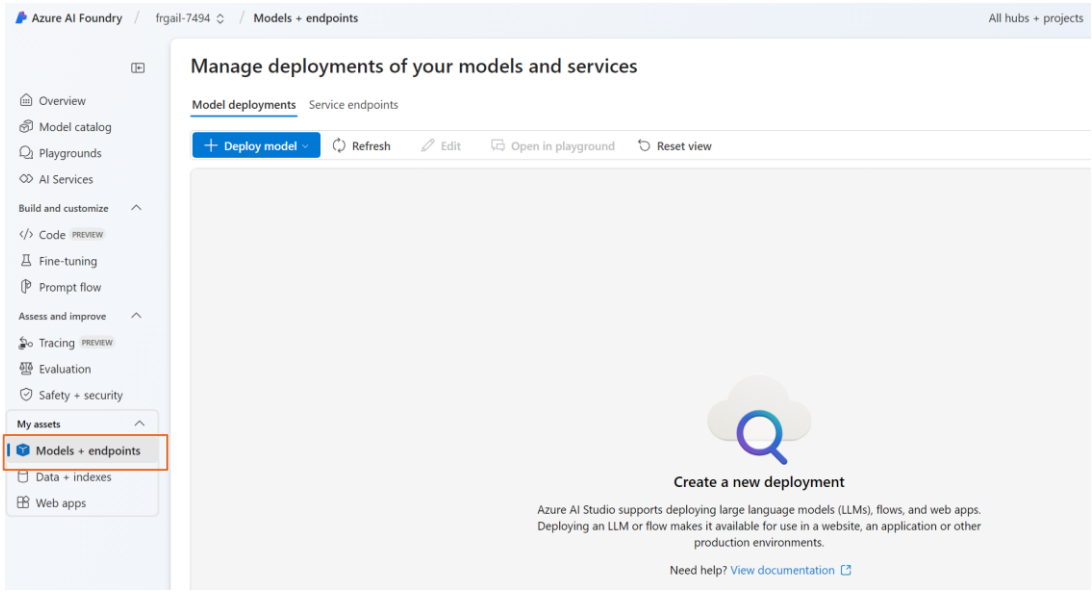
1. **Centralized Management**: A hub provides a unified environment where teams can manage security, connectivity, and computing resources across various AI projects.
2. **Collaboration**: It allows multiple developers and data scientists to collaborate on machine learning projects, sharing resources and configurations easily.
3. **Resource Sharing**: Hubs enable the sharing of Azure resources like storage accounts, model endpoints, and more, without needing repeated IT intervention.
4. **Security and Compliance**: You can set up and enforce security policies, network configurations, and compliance requirements at the hub level, which are then inherited by all projects under the hub.
5. **Project Organization**: Projects created within a hub can be customized and isolated, allowing for organized workspaces that help in managing data, access, and billing.
6. **Ease of Use**: Hubs simplify the process of setting up environments for AI development, making it easier to prototype, build, and deploy AI applications.

Deploy a model

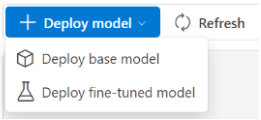
After creating your AI Project, the first step is to create a deployment of an Azure OpenAI model so you can start experimenting with the prompts you will use in your application.

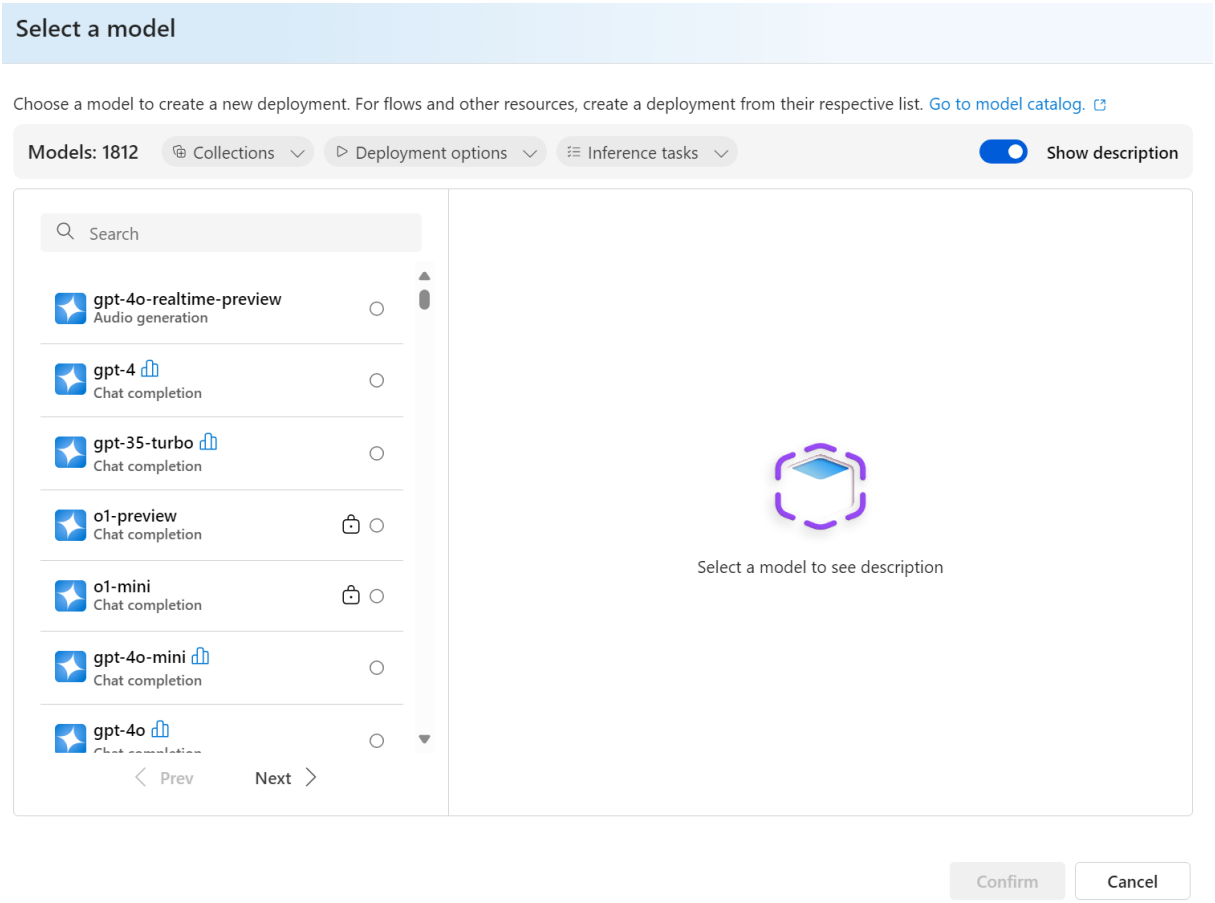
Action

- In the project you just created, click on **Models + endpoints**:

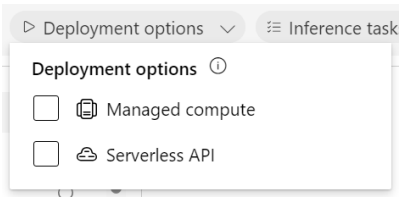


- Select Deploy base model





⚠ In Azure AI Studio, there are **two primary types of model deployment**:



1. Serverless API (MaaS : Model as a Service)

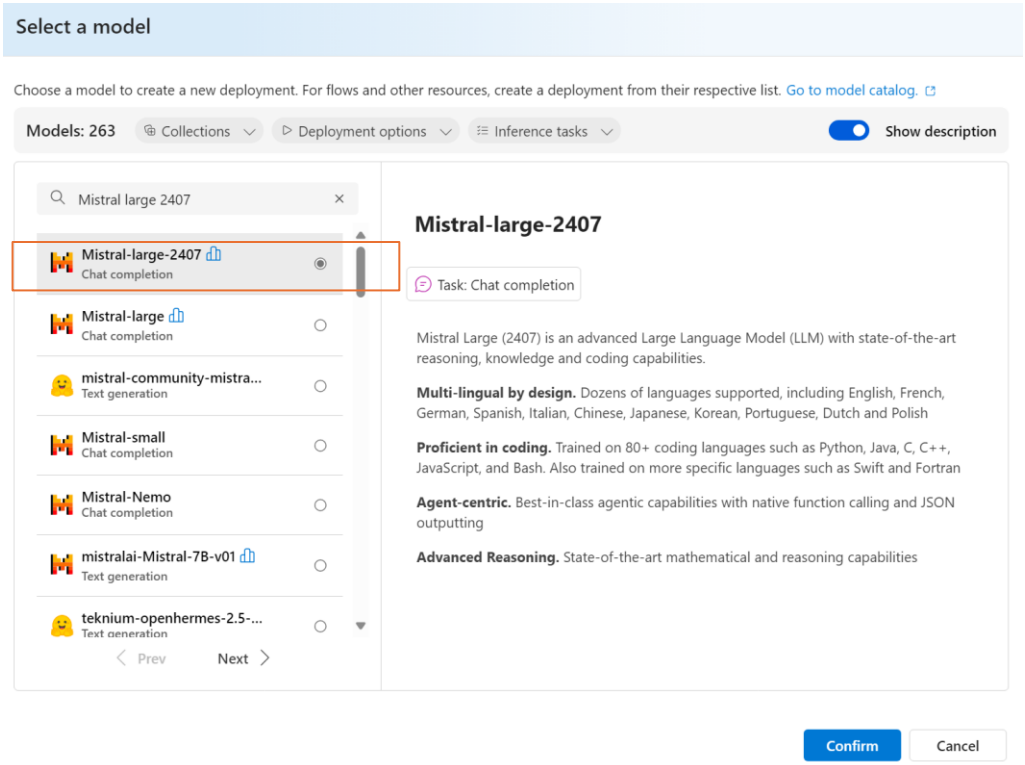
- **Description**: This deployment type allows you to deploy your model as a service without managing the underlying infrastructure.
- **Billing**: You are billed based on the number of tokens processed (pay-as-you-go).
- **Scalability**: Automatically scales to handle varying loads.
- **Use Case**: Ideal for applications where you need a flexible, cost-effective solution that can handle unpredictable traffic.

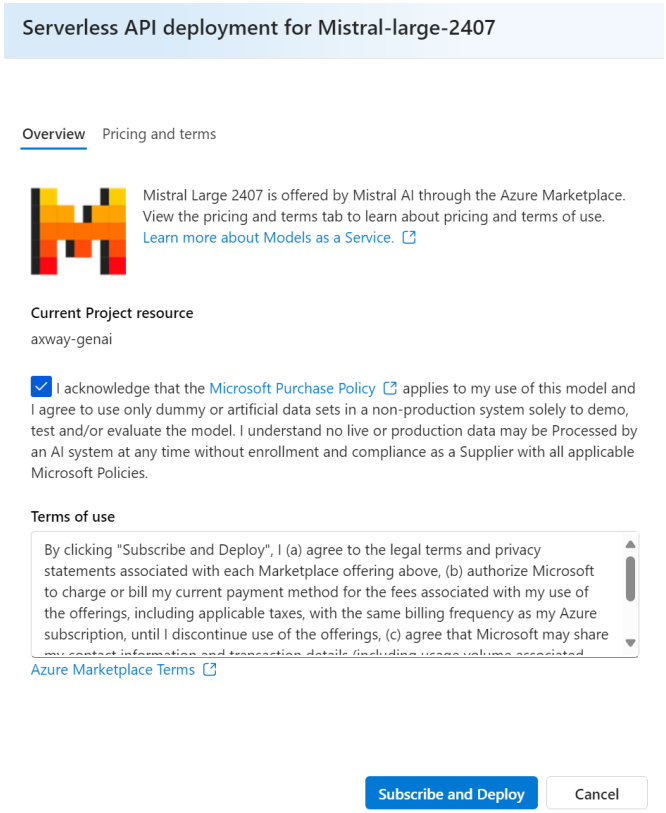
2. Managed Compute (MaaP : Model as a Platform)

- **Description**: This deployment type involves hosting your model on dedicated virtual machines within your Azure subscription.
- **Billing**: You are billed for the virtual machine core hours used.
- **Control**: Provides more control over the infrastructure, including the ability to configure the number of instances and manage capacity.

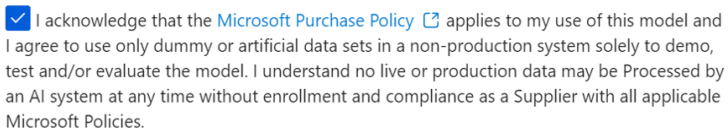
- **Use Case**: Suitable for scenarios requiring consistent performance, higher control over the environment, and potentially lower costs for predictable workloads.

- In the **search bar**, type **Mistral Large 2407** and click on **Confirm**:

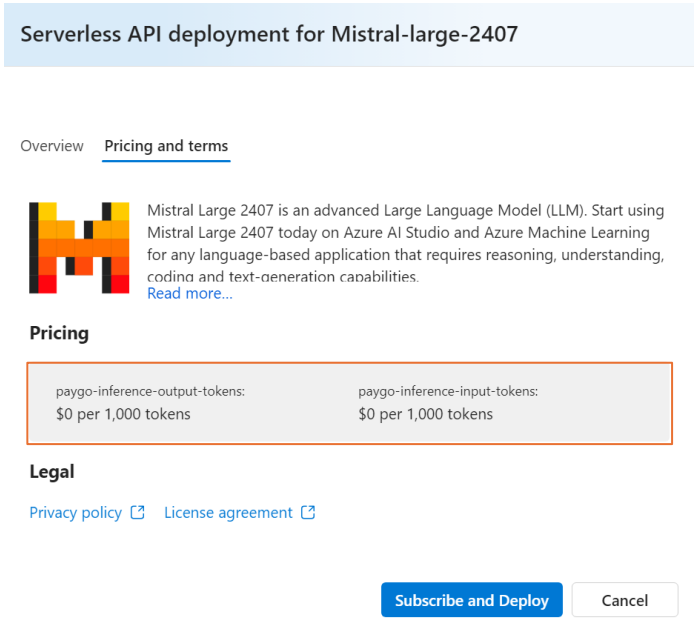




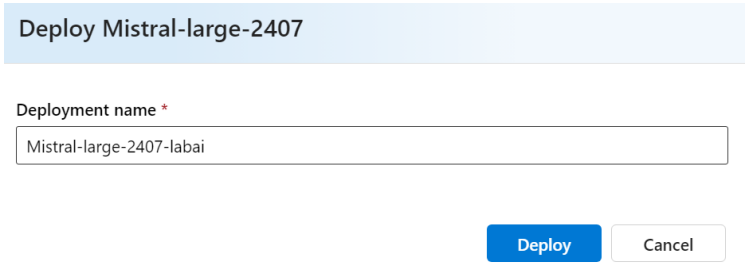
- Check the cell:

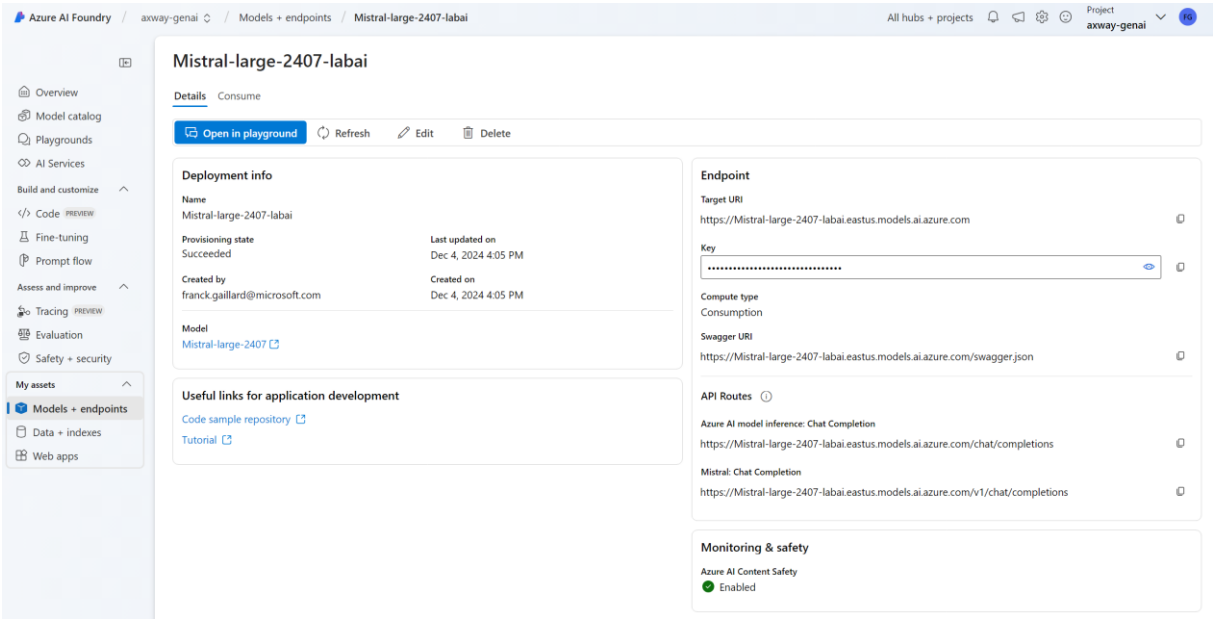


- Click on tab **"Pricing and terms"** to check the pricing:

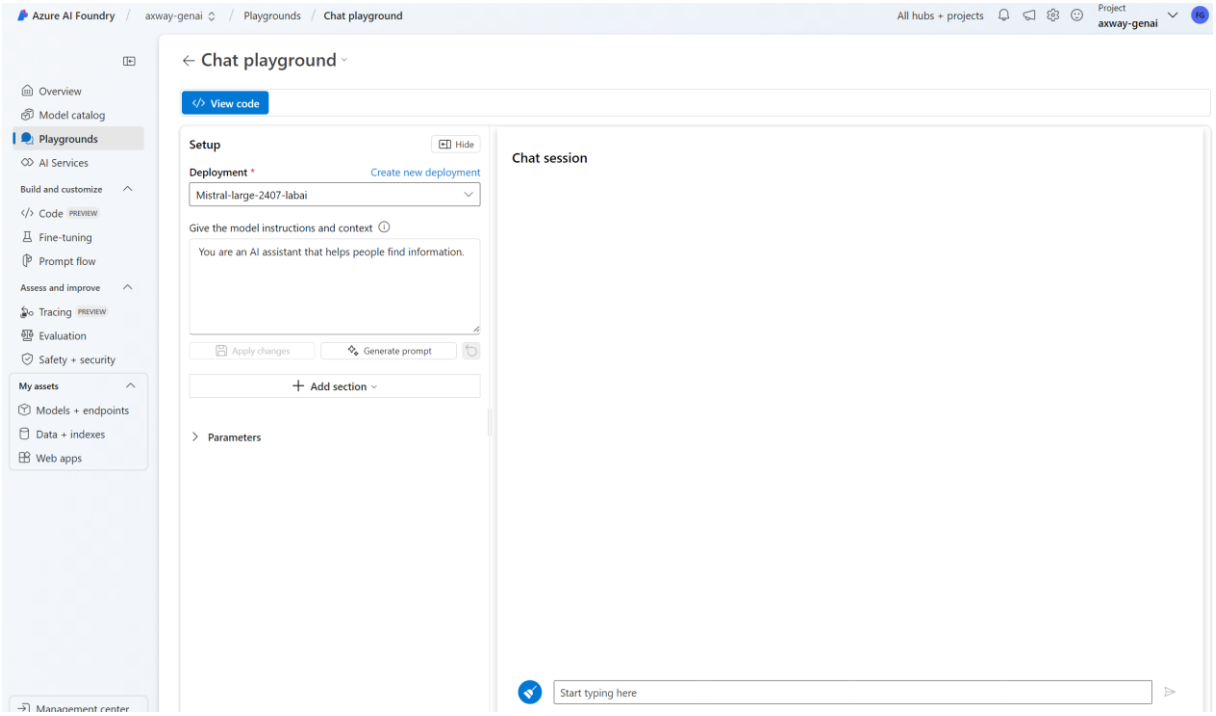


- Click on **"Subscribe and Deploy"**
- Give a name to your deployment:





- Click on **Open in playground**:



- We will **run an example** where the model will help us **summarize and extract information from a conversation** between a customer and a representative of a telco company.

  Copy the following prompt into the system message field of the playground:

  You're an AI assistant that helps telco company to extract valuable information from their conversations by creating JSON files for each conversation transcription you receive. You always try to extract and format as a JSON:

  1. Customer Name [name]

  2. Customer Contact Phone [phone]

3. Main Topic of the Conversation [topic]

4. Customer Sentiment (Neutral, Positive, Negative)[sentiment]

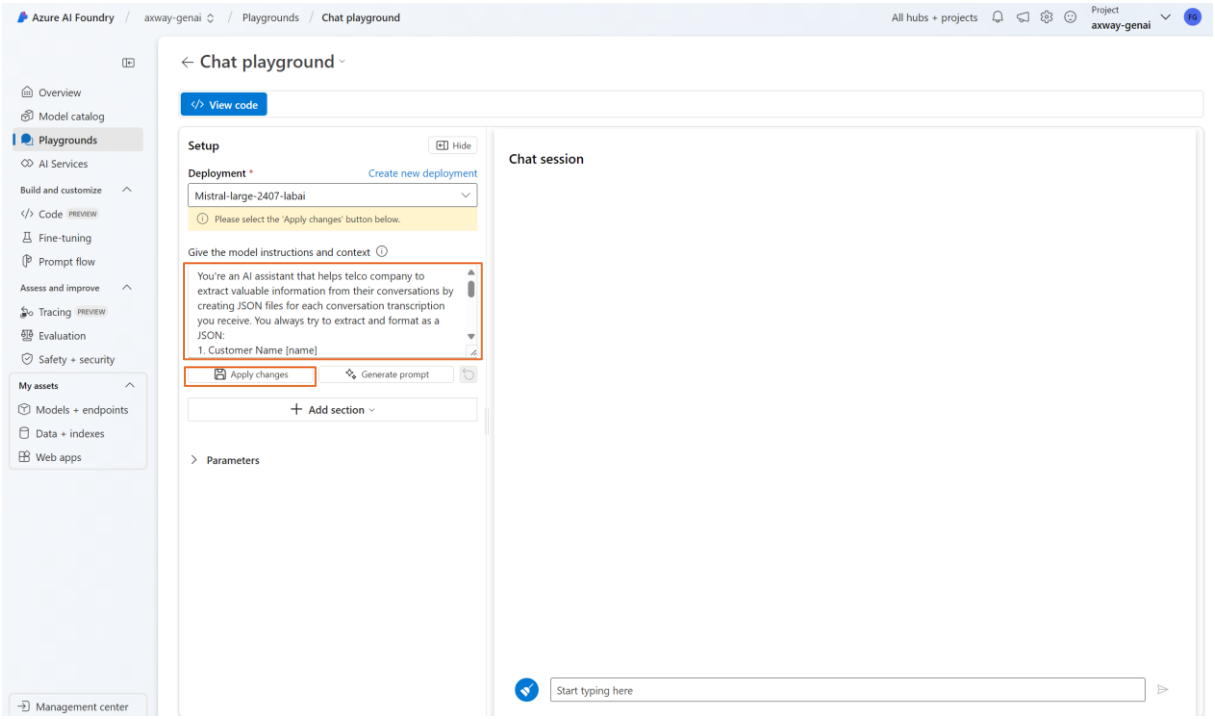
5. How the Agent Handled the Conversation [agent_behavior]

6. What was the FINAL Outcome of the Conversation [outcome]

7. A really brief Summary of the Conversation [summary]

Only extract information that you're sure. If you're unsure, write "Unknown/Not Found" in the JSON file.

After copying, select **"Apply changes"**.



Then type the following text in the chat session and click the send button:

Agent: Hello, welcome to Telco's customer service. My name is Juan, how can I assist you?

Client: Hello, Juan. I'm calling because I'm having issues with my mobile data plan. It's very slow and I can't browse the internet or use my apps.

Agent: I'm very sorry for the inconvenience, sir. Could you please tell me your phone number and your full name?

Client: Yes, sure. My number is 011-4567-8910 and my name is Martín Pérez.

Agent: Thank you, Mr. Pérez. I'm going to check your plan and your data usage. One moment, please.

Client: Okay, thank you.

Agent: Mr. Pérez, I've reviewed your plan and I see that you have contracted the basic plan of 2 GB of data per month. Is that correct?

Client: Yes, that's correct.

Agent: Well, I inform you that you have consumed 90% of your data limit and you only have 200 MB available until the end of the month. That's why your browsing speed has been reduced.

Client: What? How is that possible? I barely use the internet on my cell phone. I only check my email and my social networks from time to time. I don't watch videos or download large files.

Agent: I understand, Mr. Pérez. But keep in mind that some applications consume data in the background, without you realizing it. For example, automatic updates, backups, GPS, etc.

Client: Well, but they didn't explain that to me when I contracted the plan. They told me that with 2 GB I would have enough for the whole month. I feel cheated.

Agent: I apologize, Mr. Pérez. It was not our intention to deceive you. I offer you a solution: if you want, you can change your plan to a higher one, with more GB of data and higher speed. This way you can enjoy a better browsing experience.

Client: And how much would that cost me?

Agent: We have a special offer for you. For only 10 pesos more per month, you can access the premium plan of 5 GB of data and 4G speed. Are you interested?

Client: Mmm, I don't know. Isn't there another option? Can't you give me more speed without charging me more?

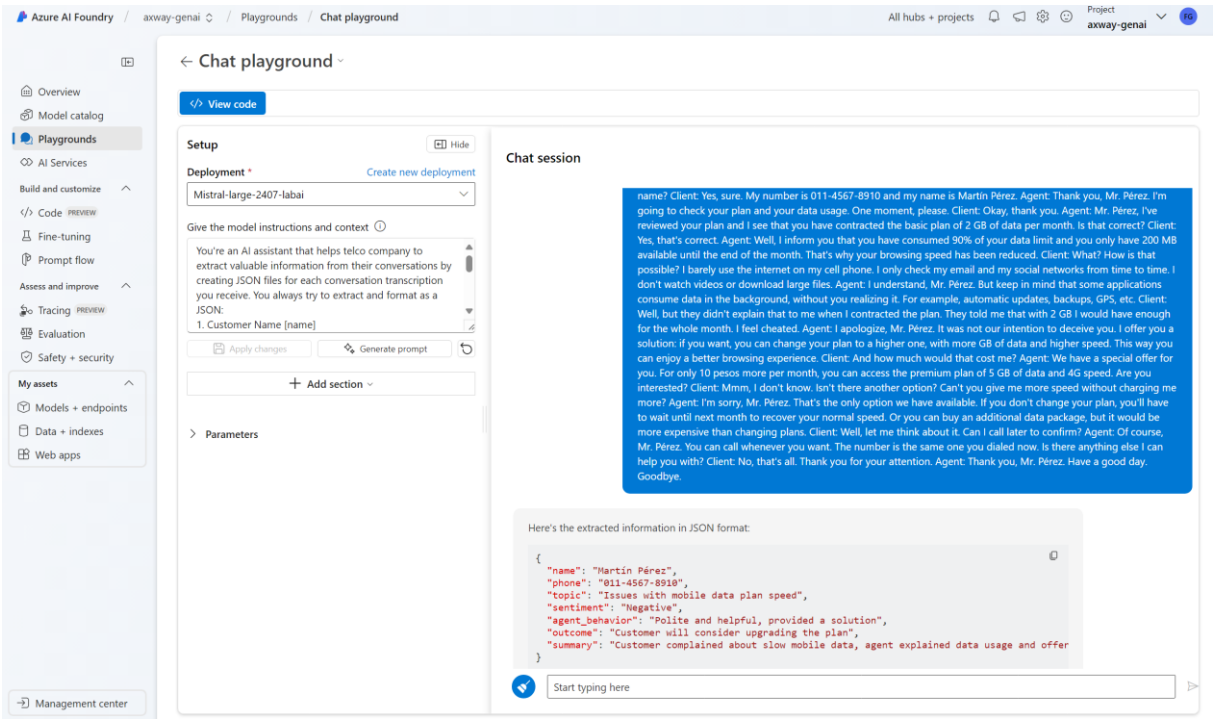
Agent: I'm sorry, Mr. Pérez. That's the only option we have available. If you don't change your plan, you'll have to wait until next month to recover your normal speed. Or you can buy an additional data package, but it would be more expensive than changing plans.

Client: Well, let me think about it. Can I call later to confirm?

Agent: Of course, Mr. Pérez. You can call whenever you want. The number is the same one you dialed now. Is there anything else I can help you with?

Client: No, that's all. Thank you for your attention.

Agent: Thank you, Mr. Pérez. Have a good day. Goodbye.



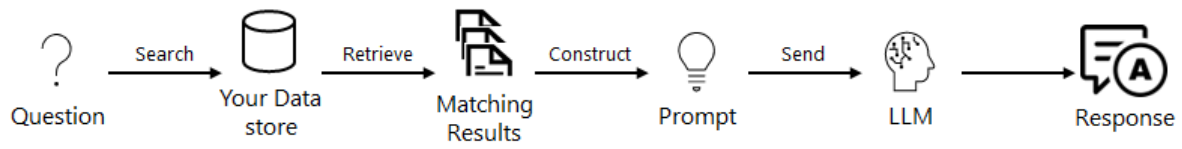
Retrieval Augmented Generation (RAG)

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 :



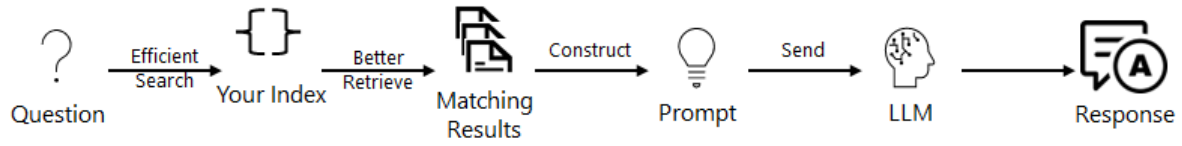
What is an index and why do I need it?

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:



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.

Action

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 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

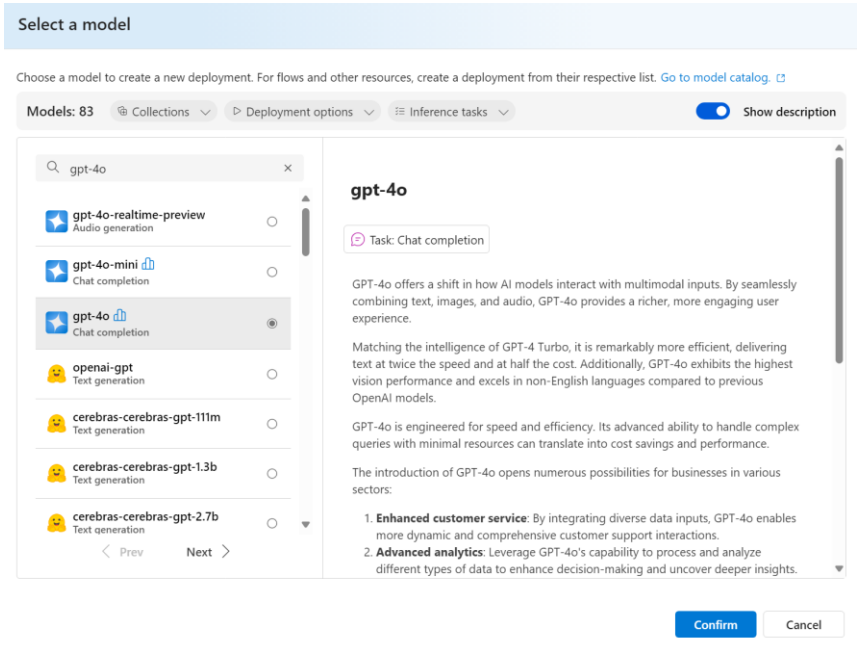
- You need to have Microsoft.Web resource provider registered in the selected subscription, to be able to deploy to a web app. To verify:
    - **Sign in to the Azure Portal**: Go to the Azure Portal and log in with your credentials.
    - **Navigate to Subscriptions**:
        - In the Azure Portal 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

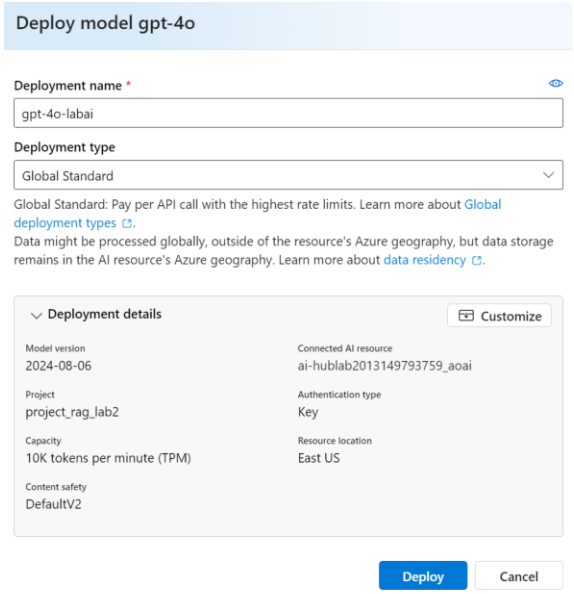
Follow these steps to add your data in the chat playground to help the assistant answer questions about your products. You're not changing the deployed model itself. Your data is stored separately and securely in your Azure subscription.

Deploy a GPT-4o model

1. Open your project
2. Click on Models + endpoints
3. Click on Deploy model / Deploy base model
4. In the **search bar**, type **GPT-4o**, select **gpt-4o** and click on **Confirm**:

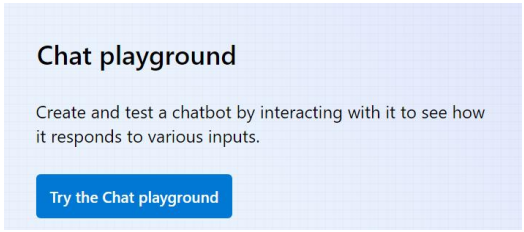


5. Click on **Deploy**:

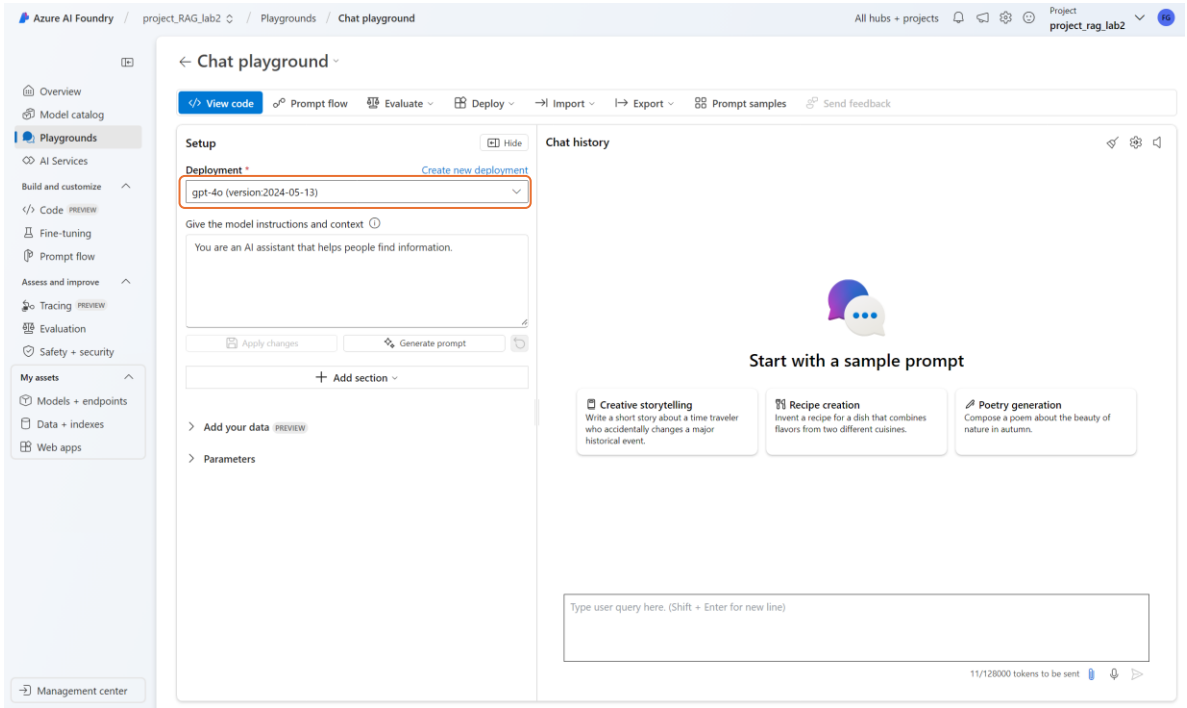


Deploy a RAG model

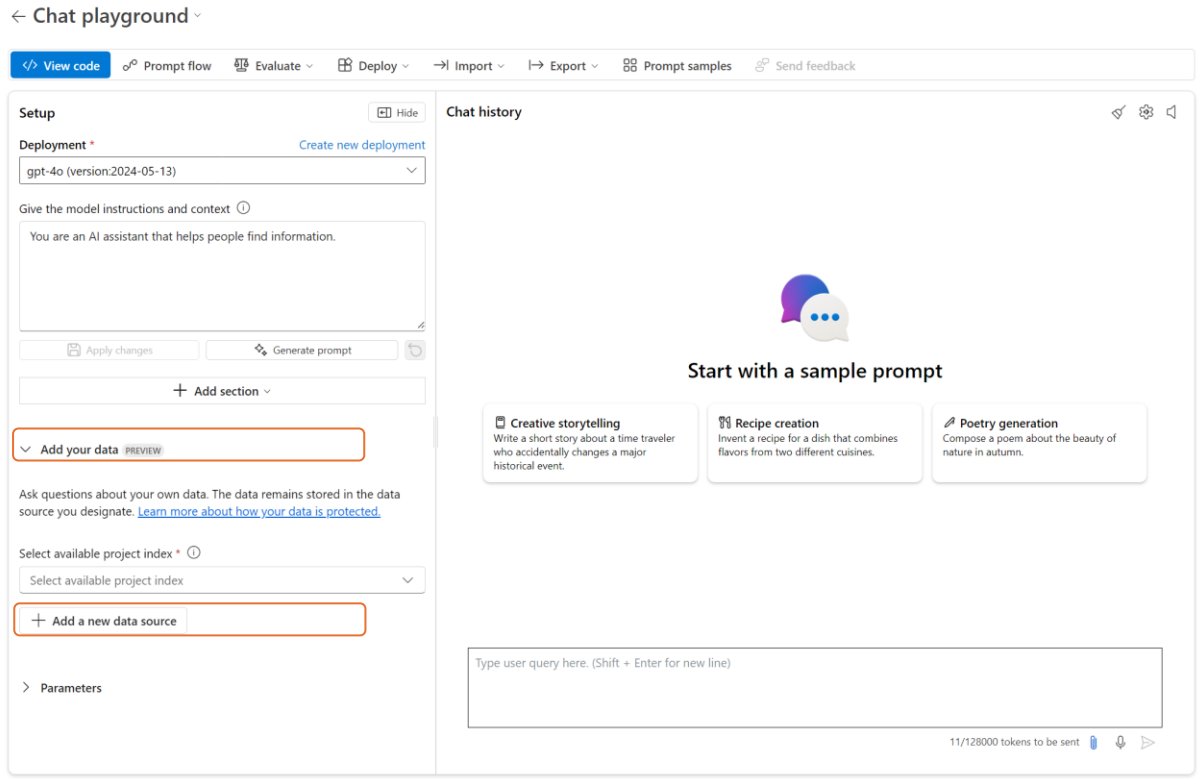
1. Select **Playgrounds**, and click on **Try the Chat playground**:

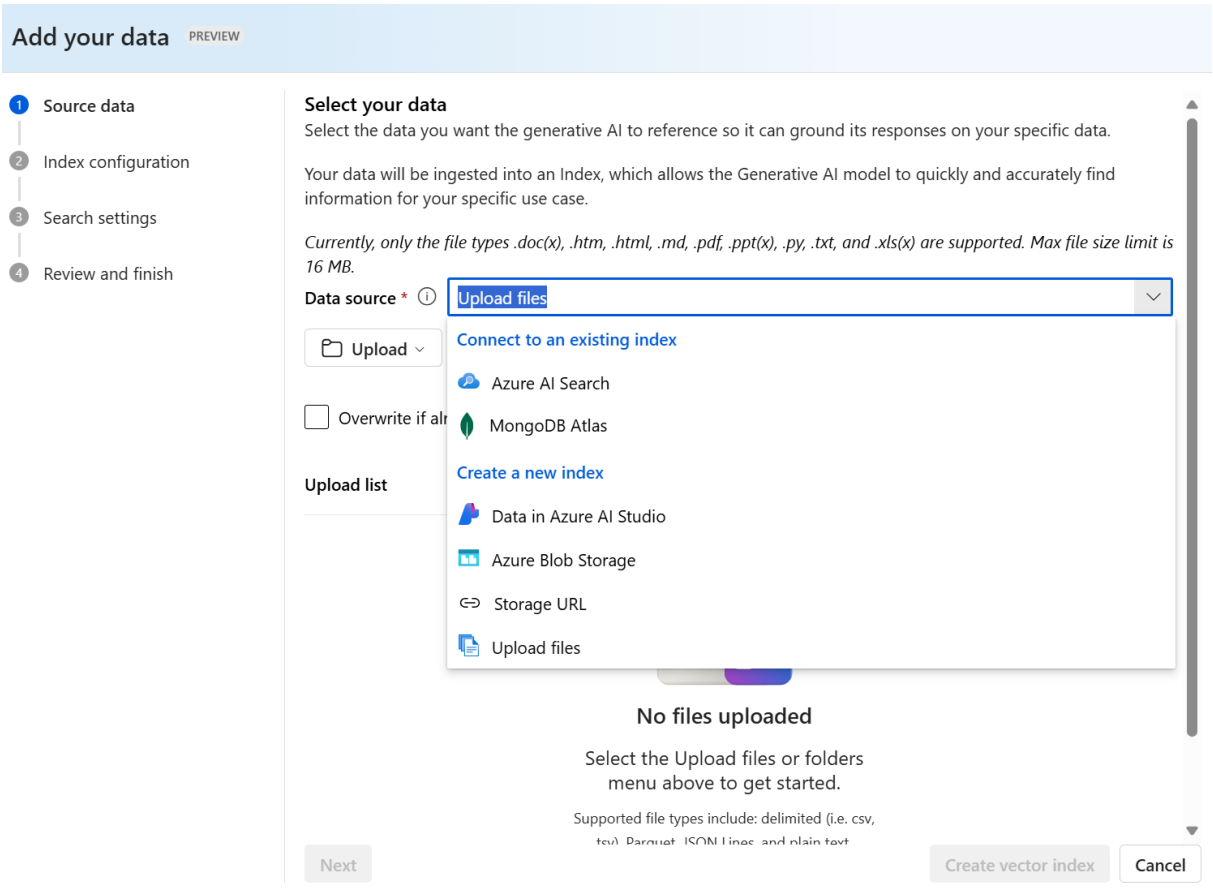


2. Select your deployed chat model from the **Deployment** dropdown.

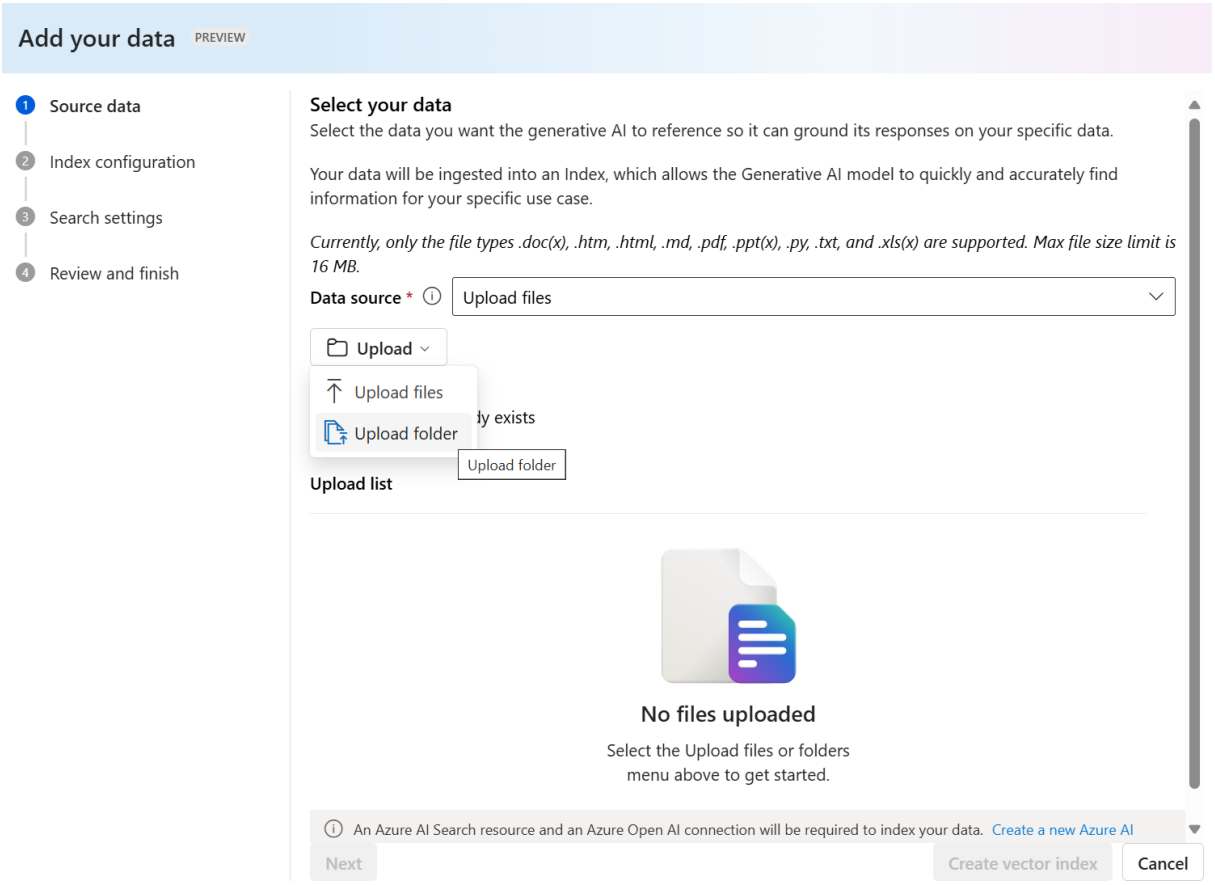


3. On the left side of the chat playground, select **Add your data** > **+ Add a new data source**.

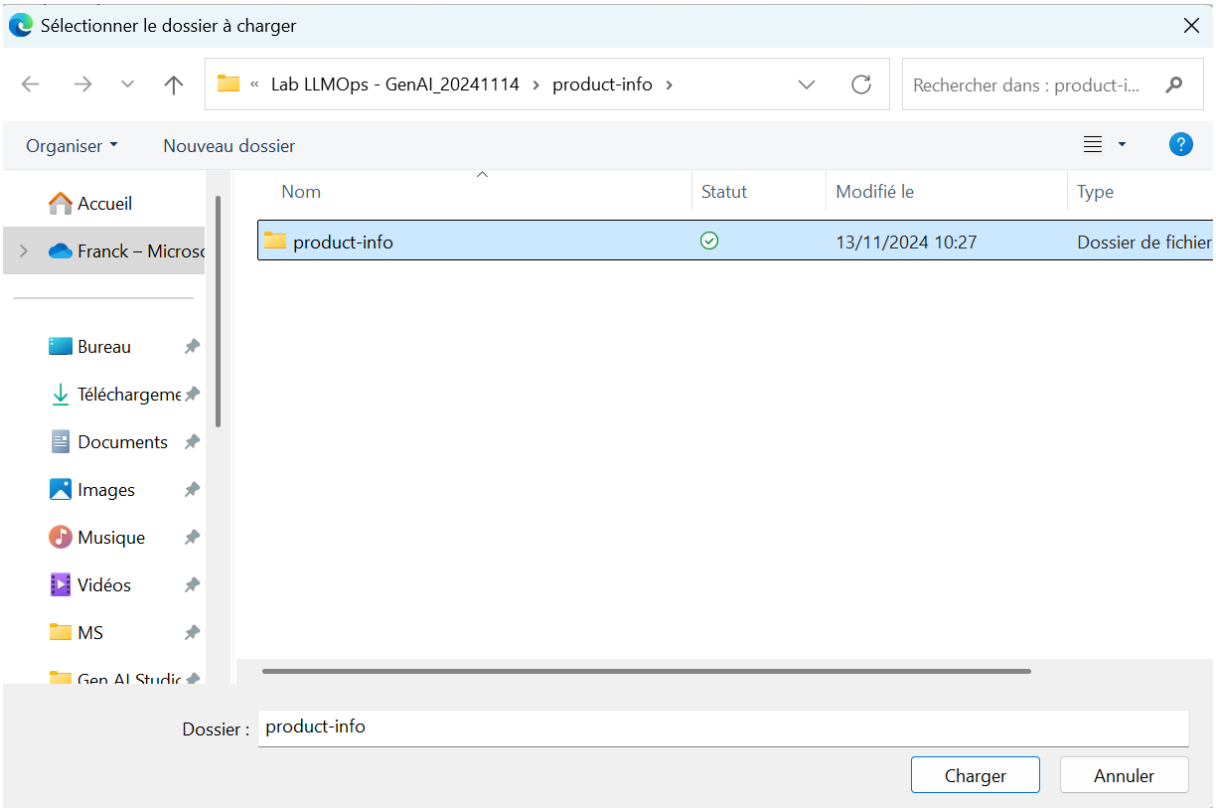


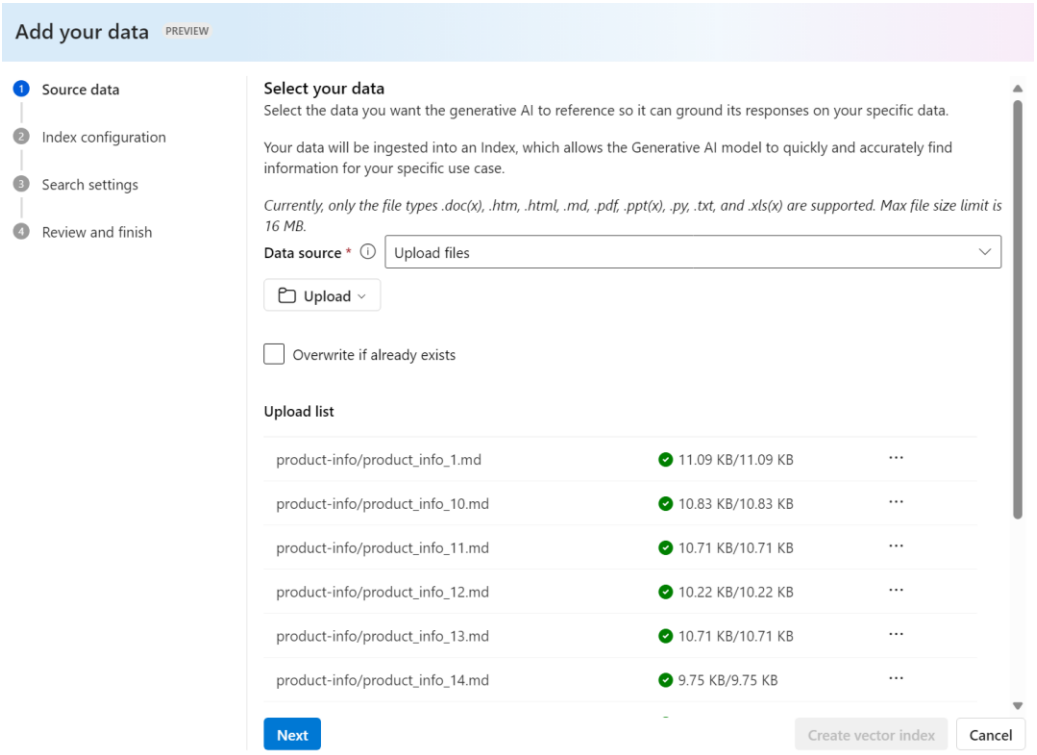


4. Click on “**Upload files**” and “**Upload folder**”:

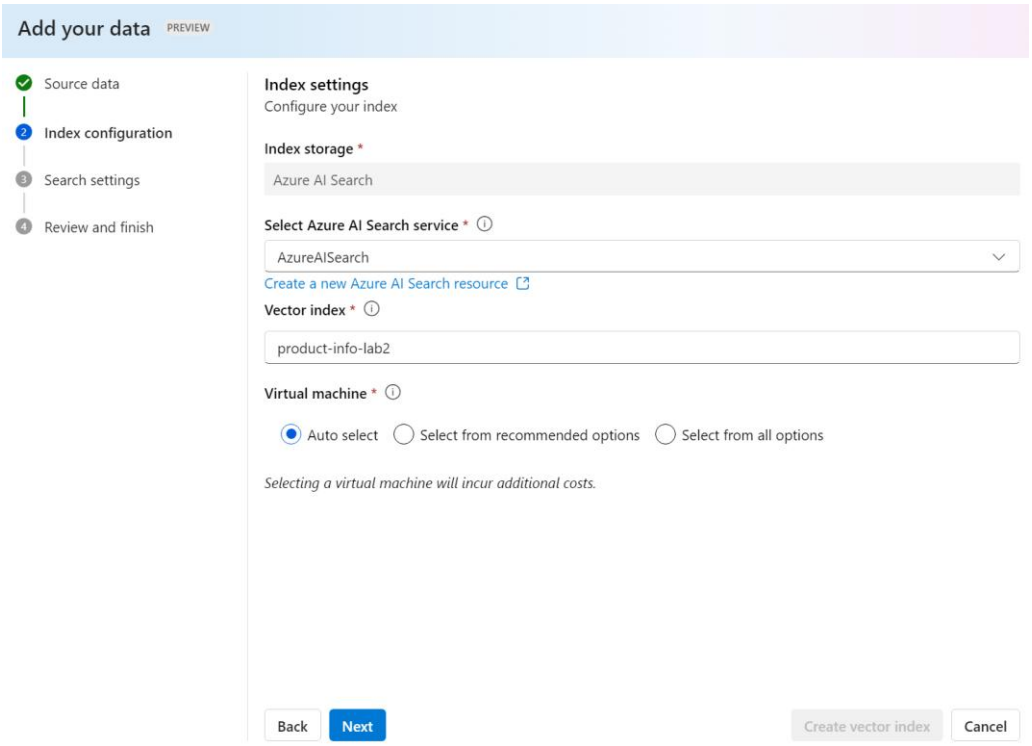


5. Select "**product-info**" folder. The Markdown files are uploaded to the blob storage.

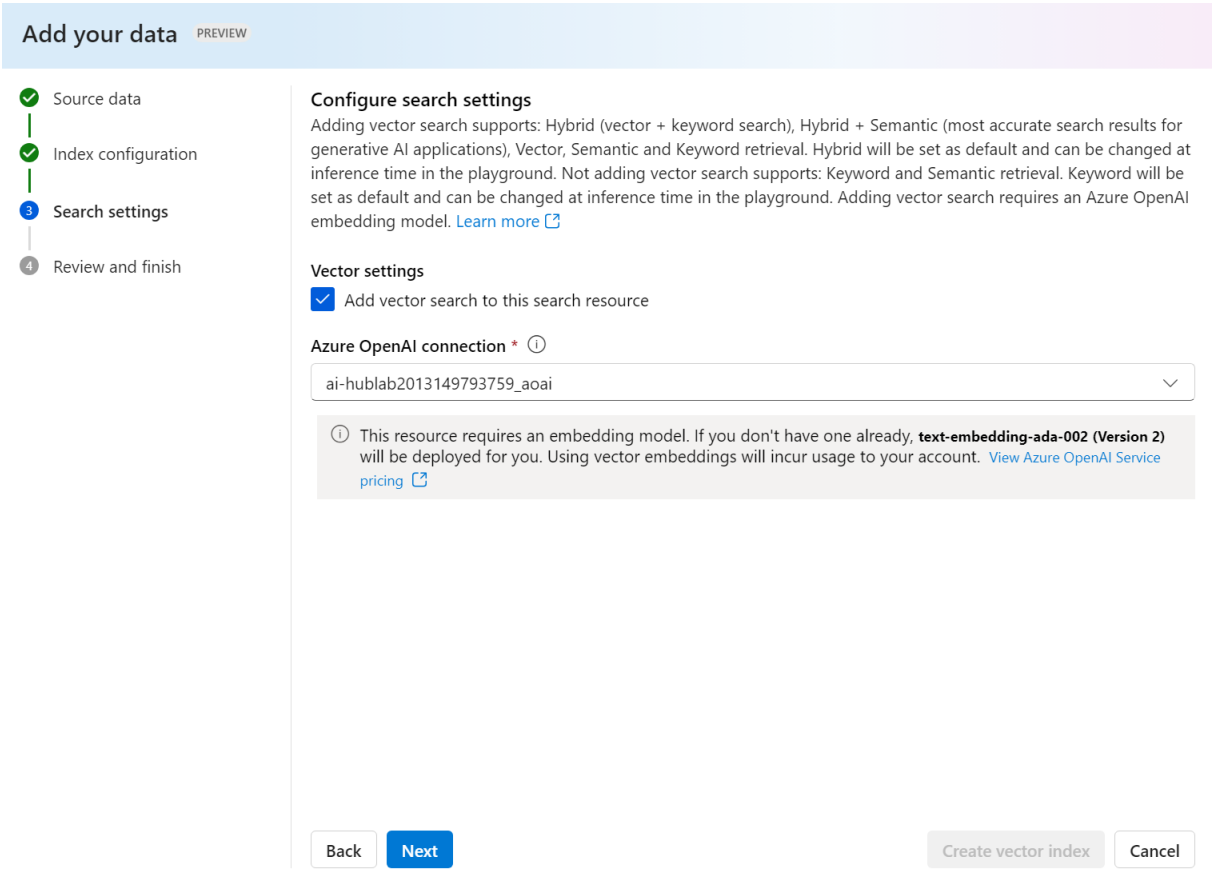




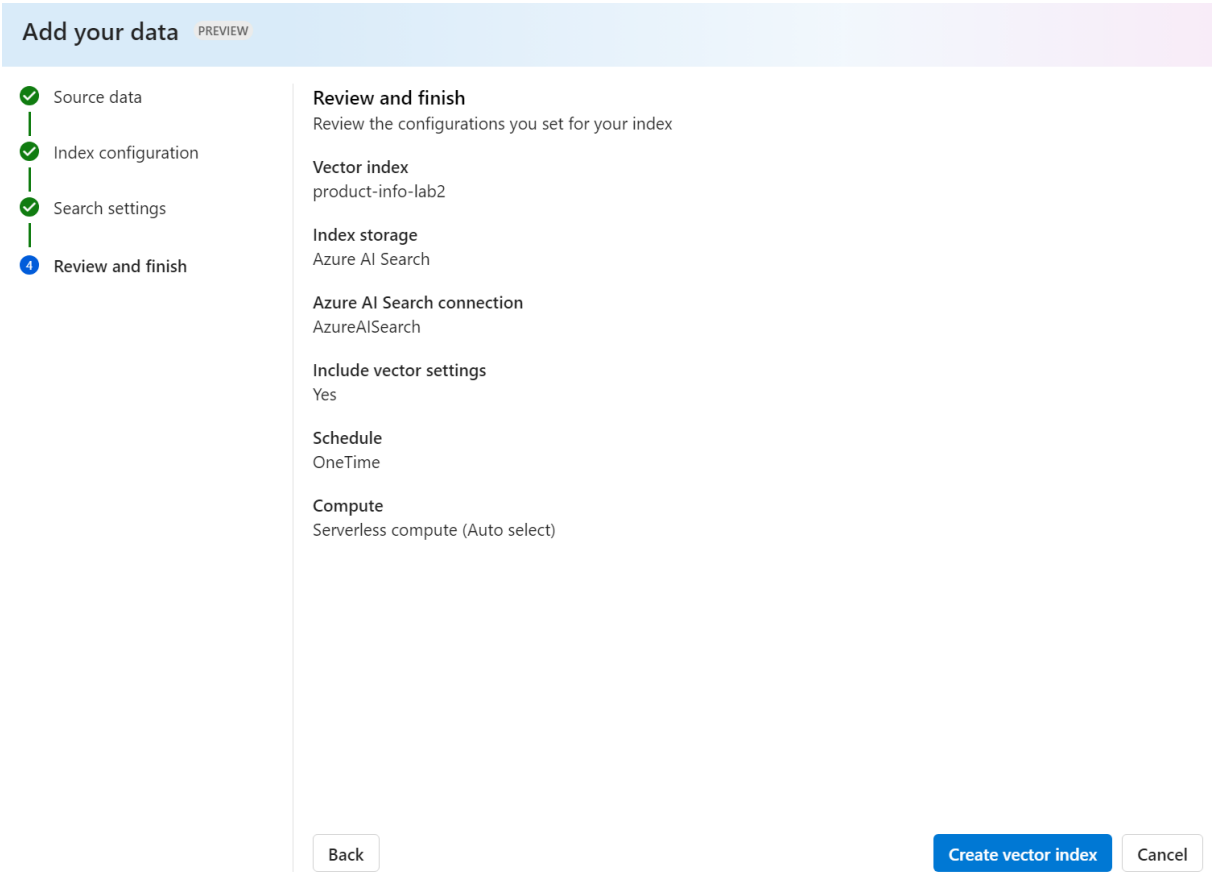
6. Select the Azure AI Search service from the drop-down menu and give your index a name. Leave “Auto select” for the virtual machine.



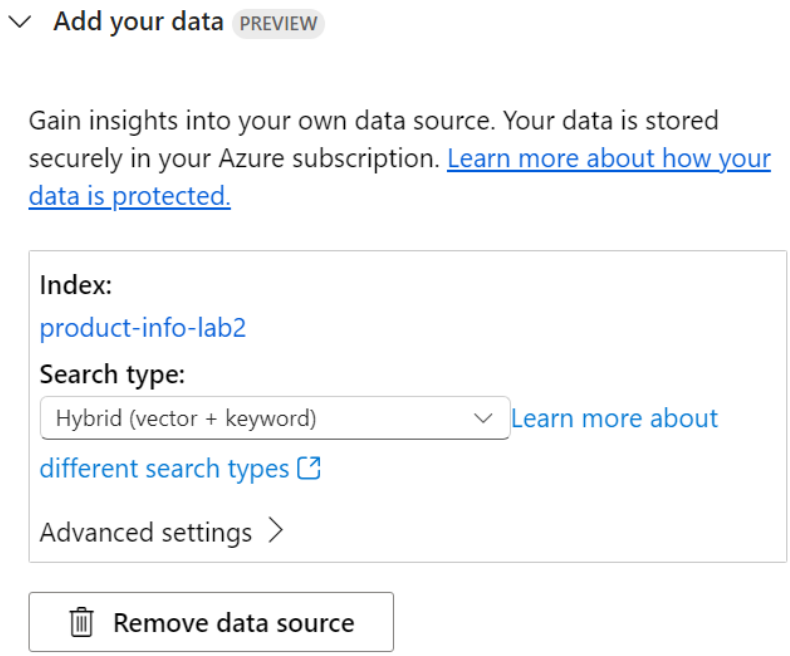
7. Click on “**Add vector search to this search resource**” and select an **Azure OpenAI connection** from the drop-down menu:



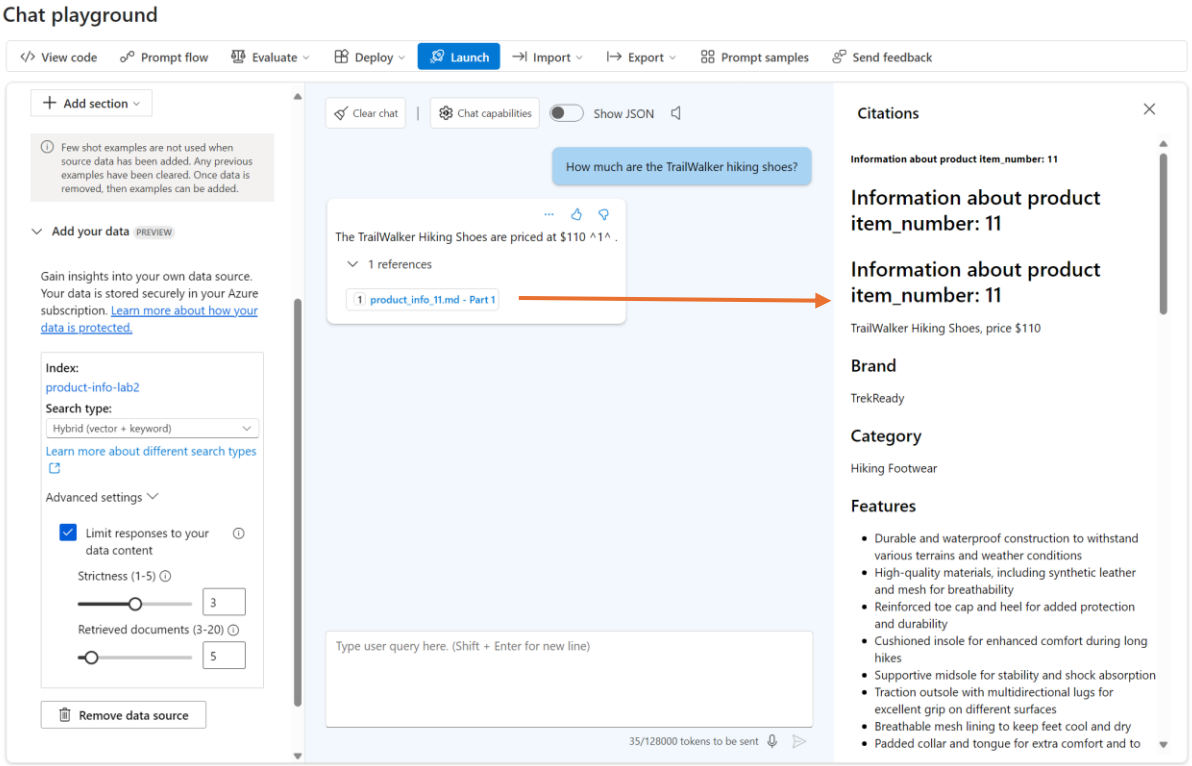
8. Click on **Create vector index**:



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



10. 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.

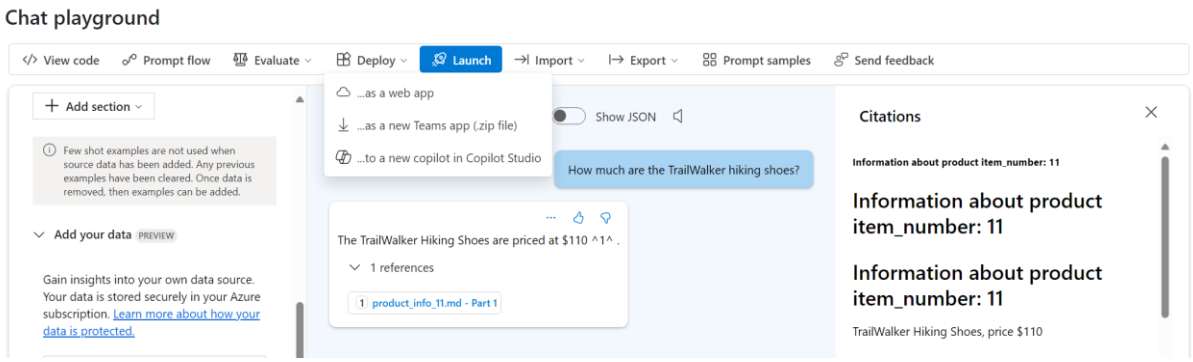


Deploy your web app

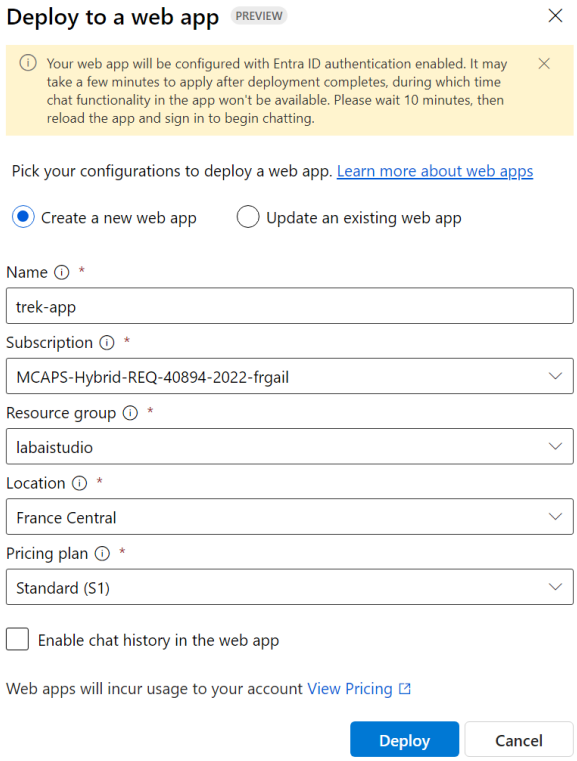
Once you're satisfied with the experience in Azure AI Studio, you can deploy the model as a standalone web application. Publishing creates an Azure App Service in your subscription. It

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**":



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



3. 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, with its own SDK and VS Code extension.

Benefits of prompt flow:

With prompt flow in Azure AI Studio, 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.

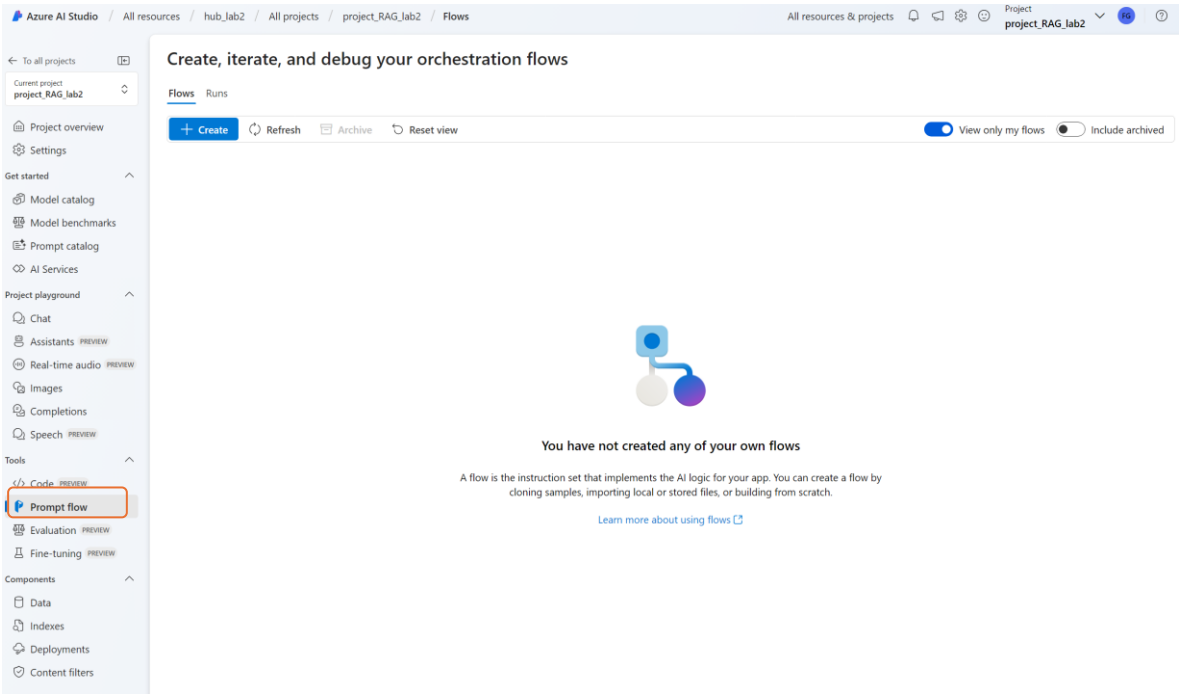
Enterprise readiness :

- *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.

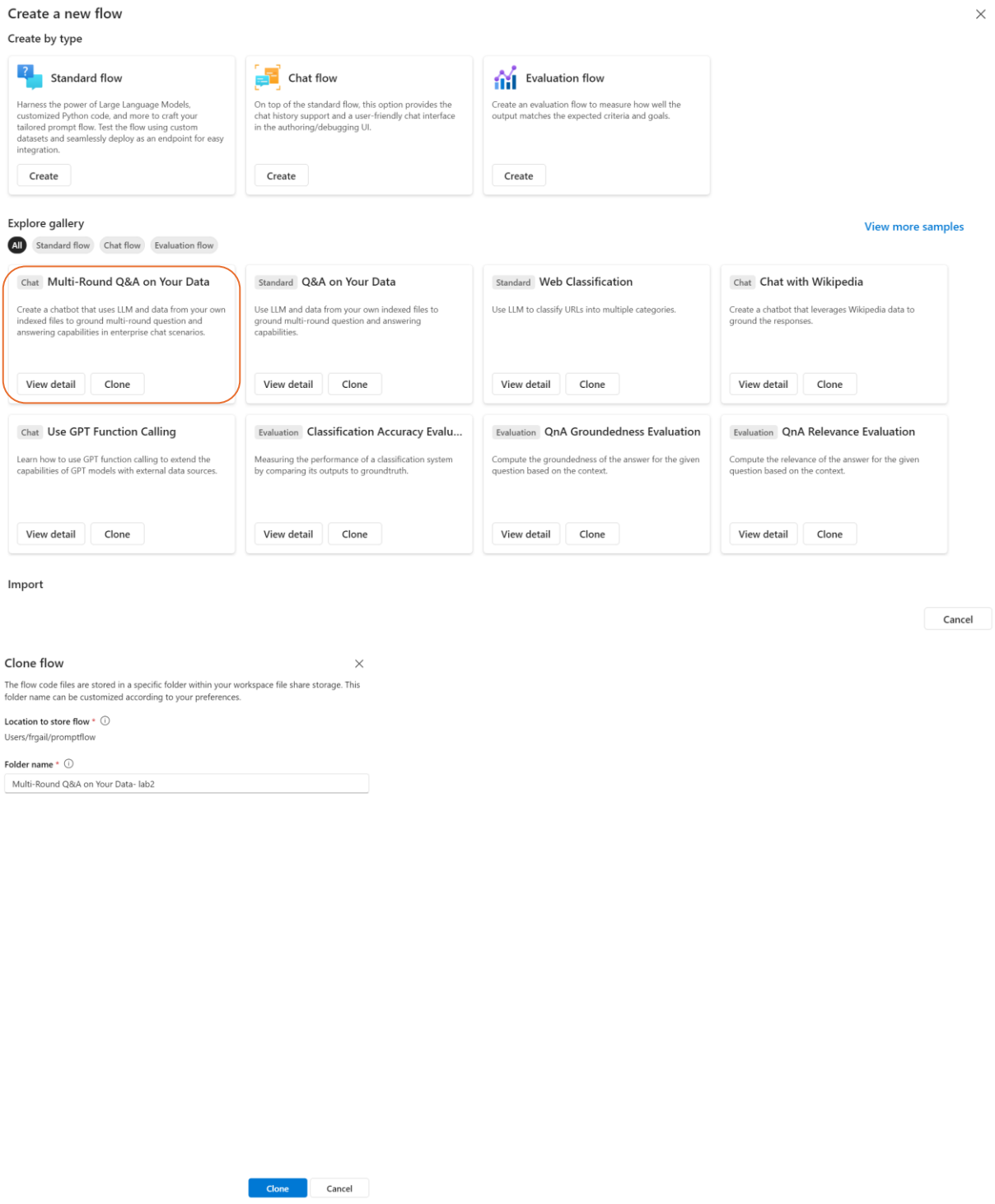
With prompt flow in Azure AI Studio, you can unleash prompt engineering agility, collaborate effectively, and apply enterprise-grade solutions for successful LLM-based application development and deployment.

Action

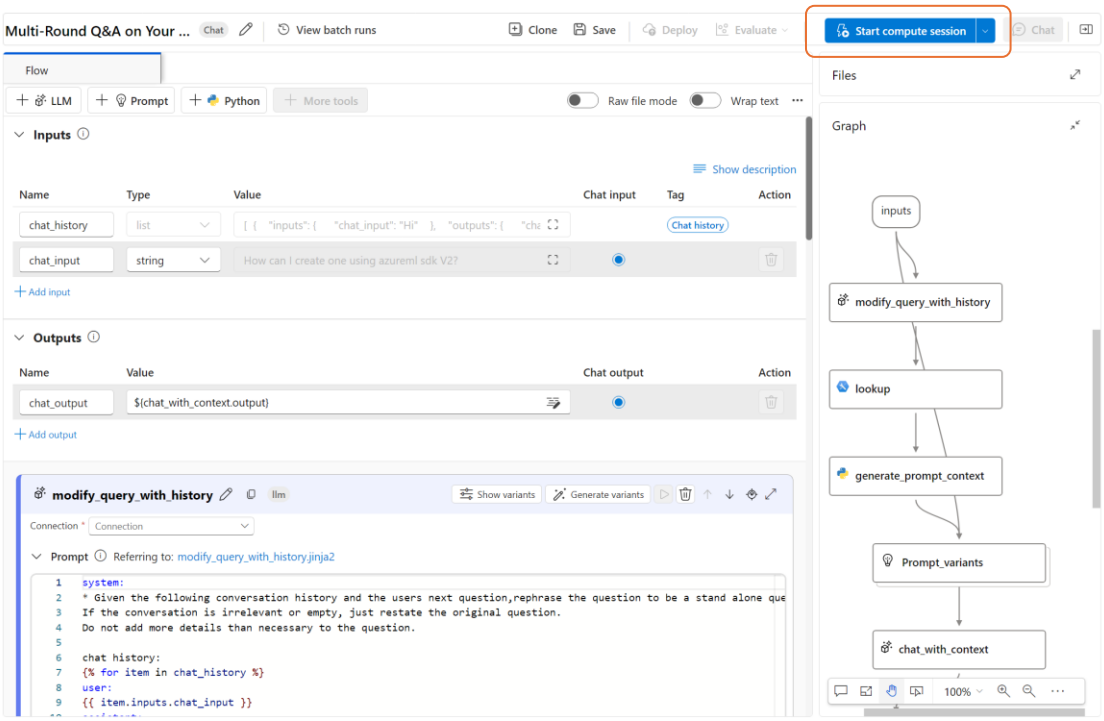
1. Click on **Prompt Flow**:

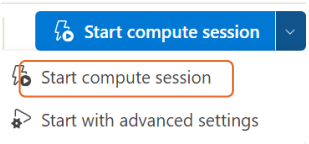


2. Click on **Create** and **Clone “Multi-Round Q&A on Your Data”**:



3. 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





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.

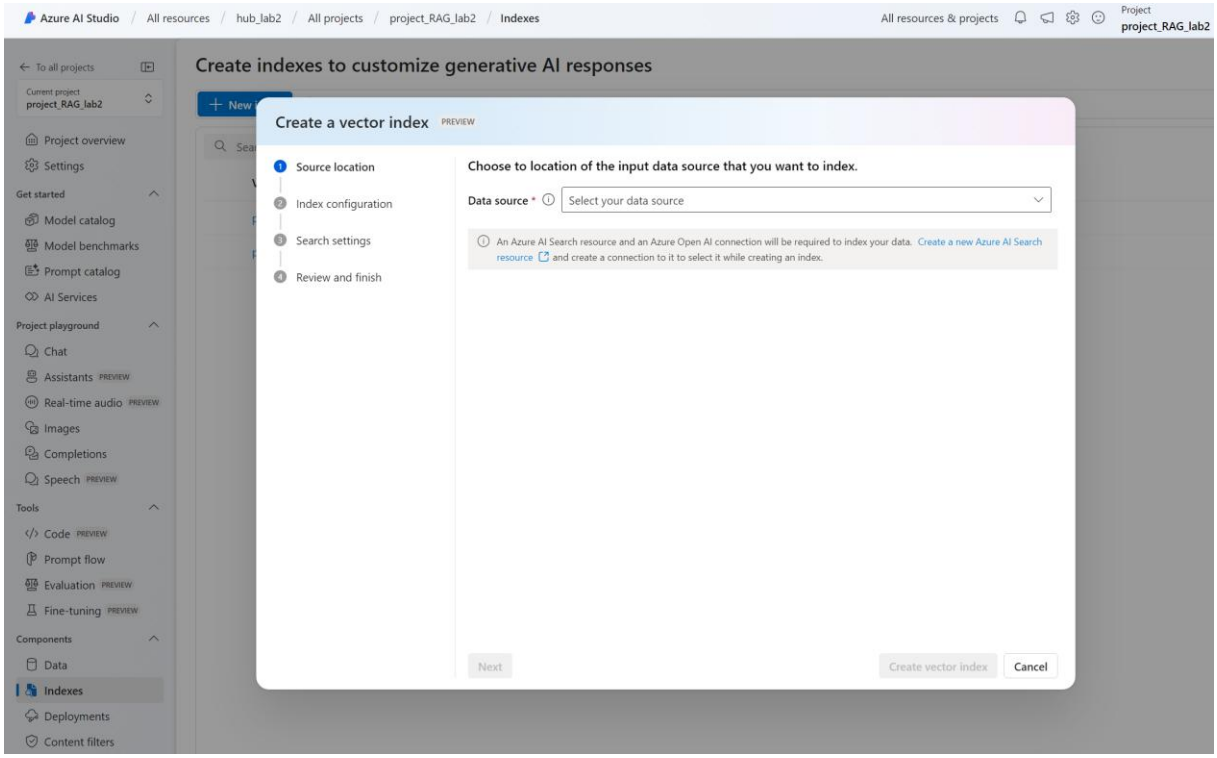
This string then serves as input for the **Prompt_variants** node, which formulates various prompts.

Finally, these prompts are used to generate the user's answer in the **chat_with_context** node.

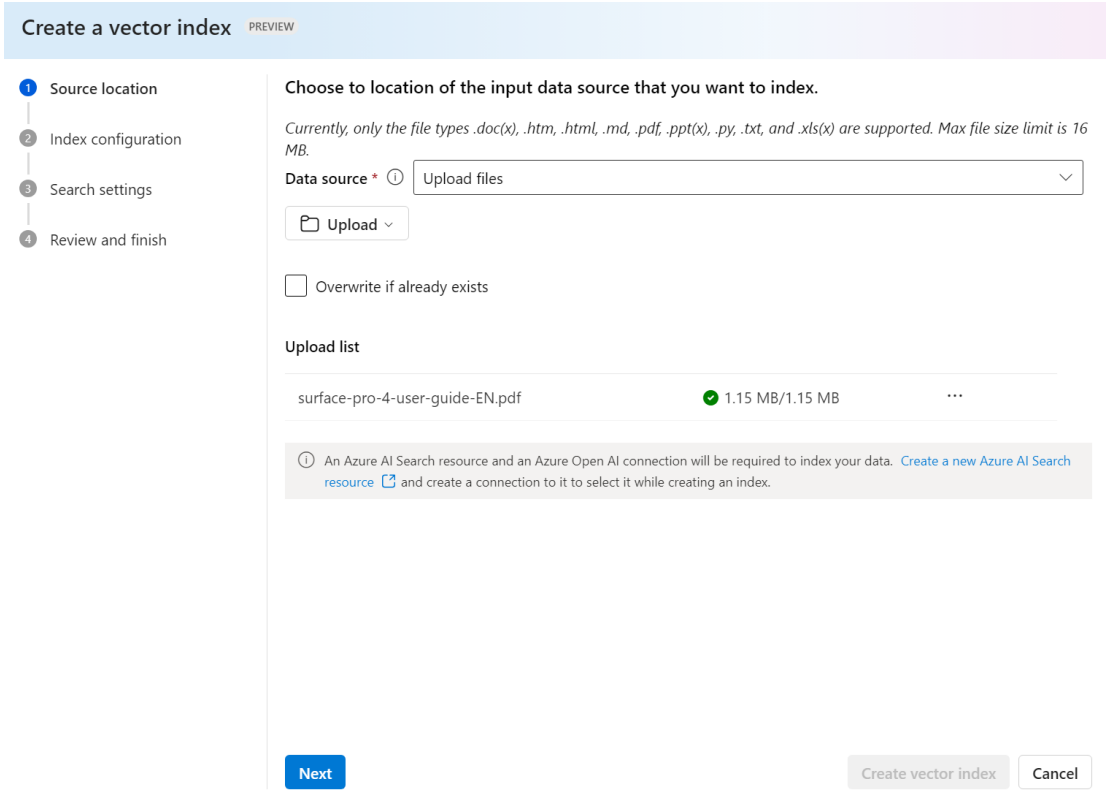
4. Create a search index

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.

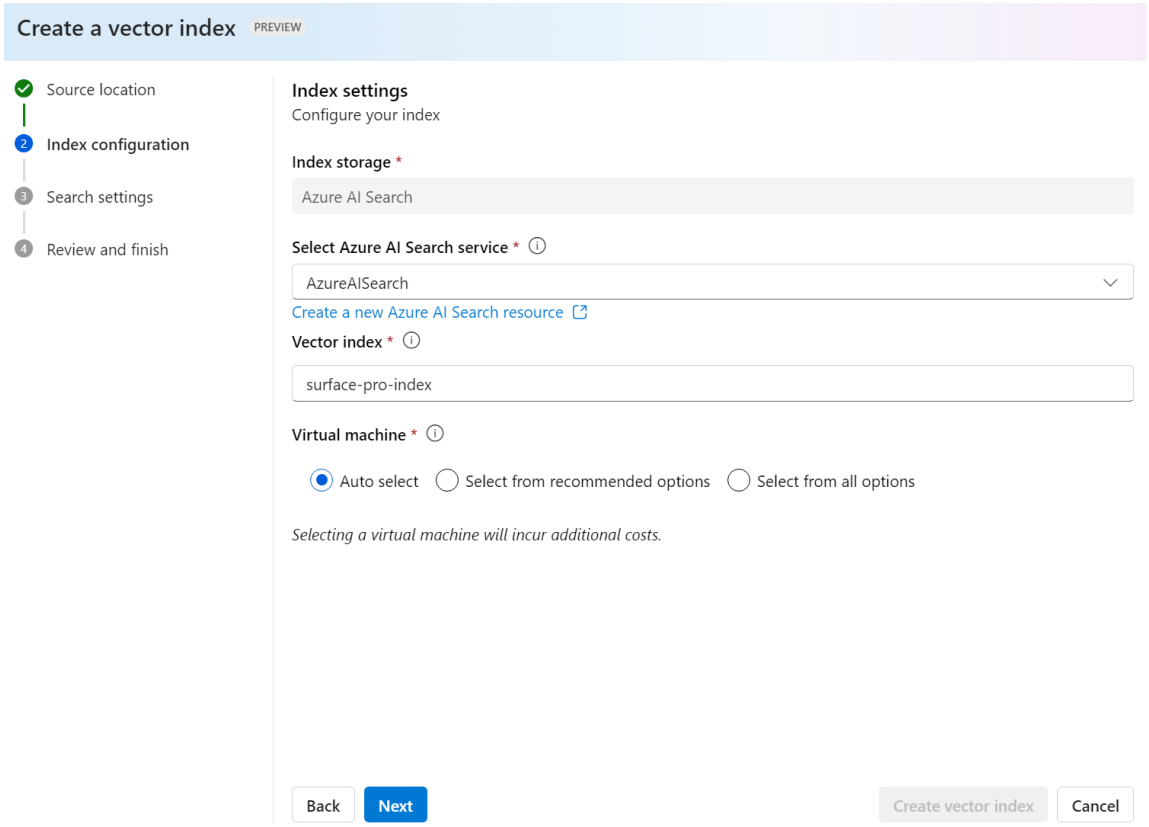
In our case, we will create a **Vector index**. To do this, you just need to go back to the project in the **AI Studio**, select the **Indexes** option, and then click on the **New index** button.

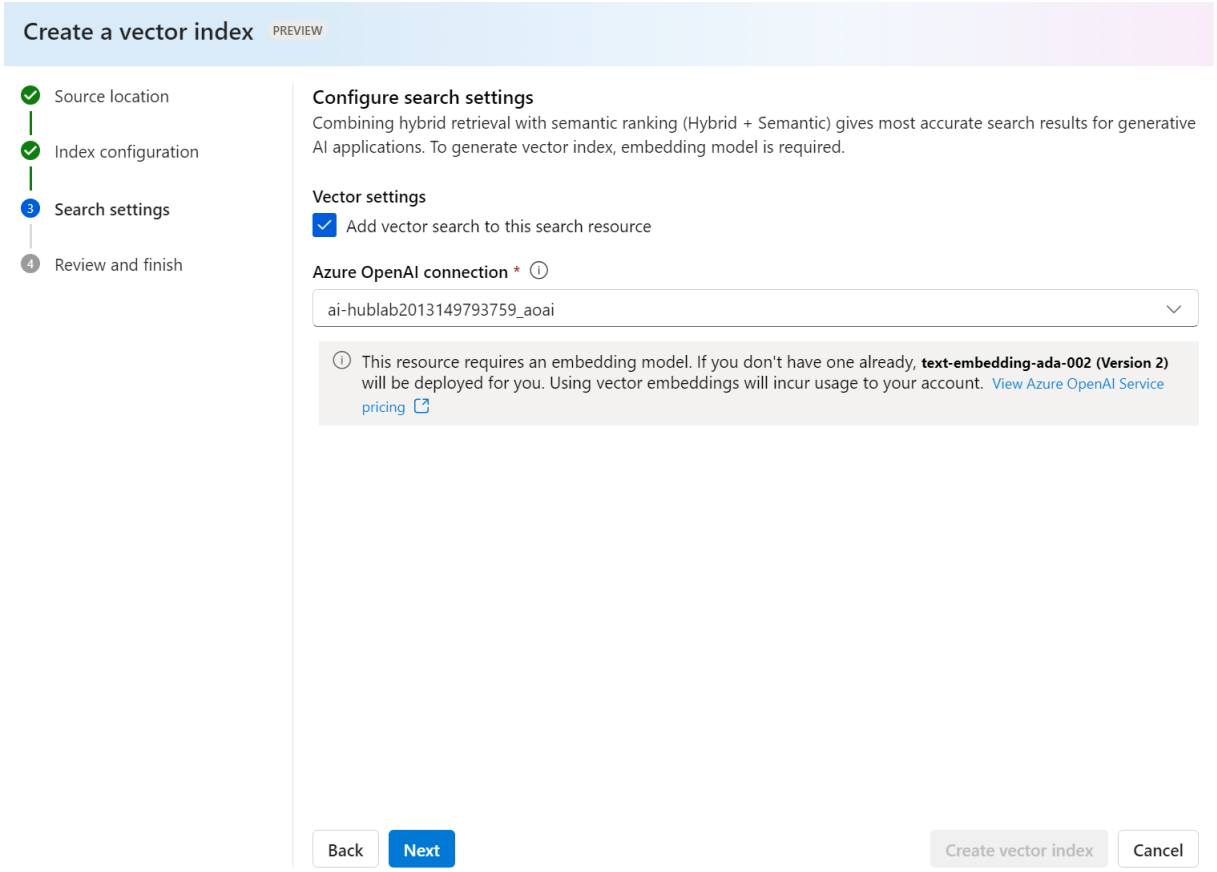


At the **Data source** stage, select the **Upload files/folders** option and upload the **PDF files/surface-pro-4-user-guide-EN.pdf** to the data folder of this lab, as shown in the next screen.

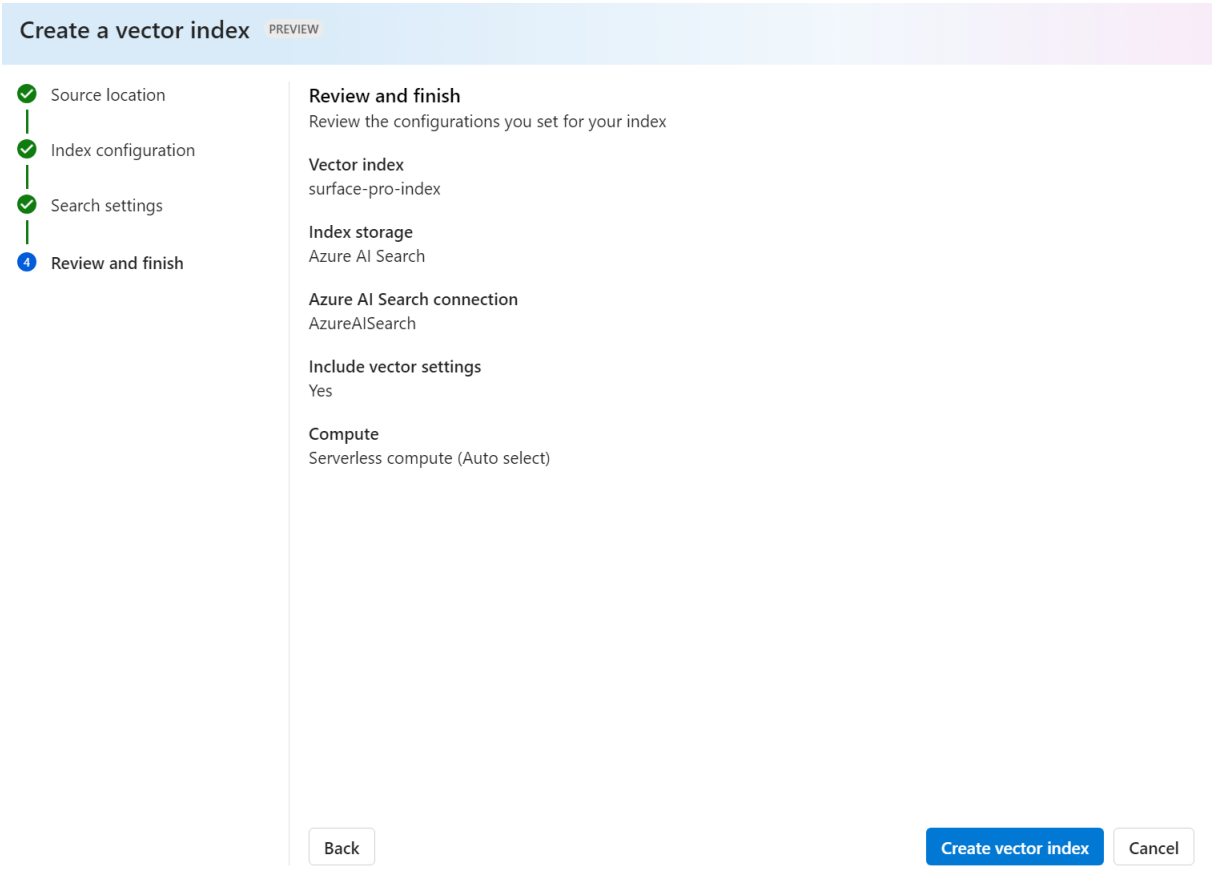


In **Index storage**, select the Search Service you created earlier, give a name to your index and select "**Auto select**" for the virtual machine.

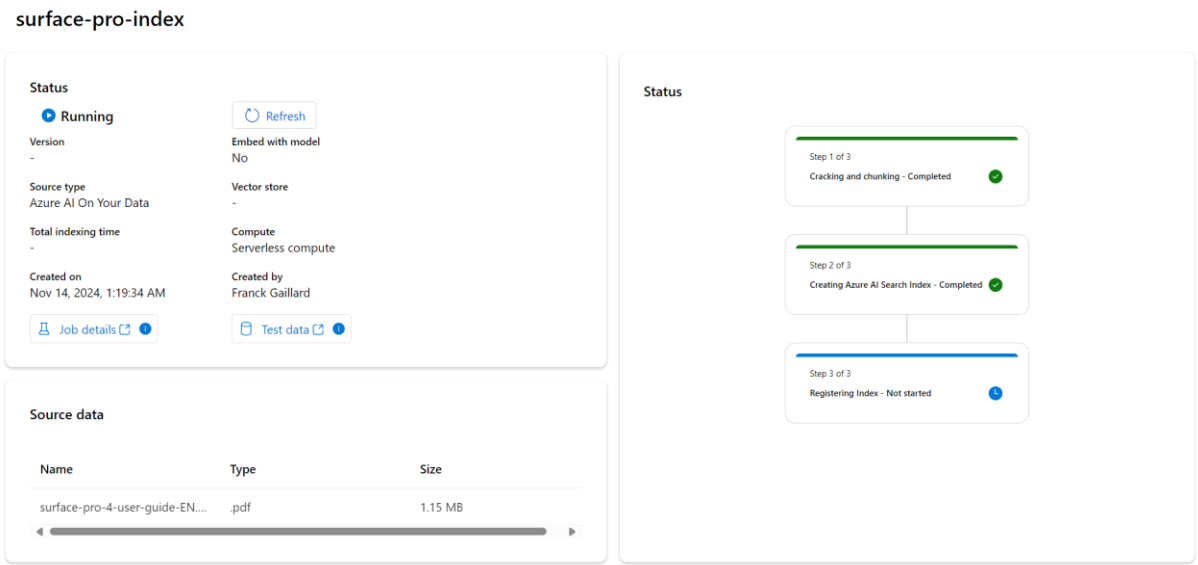




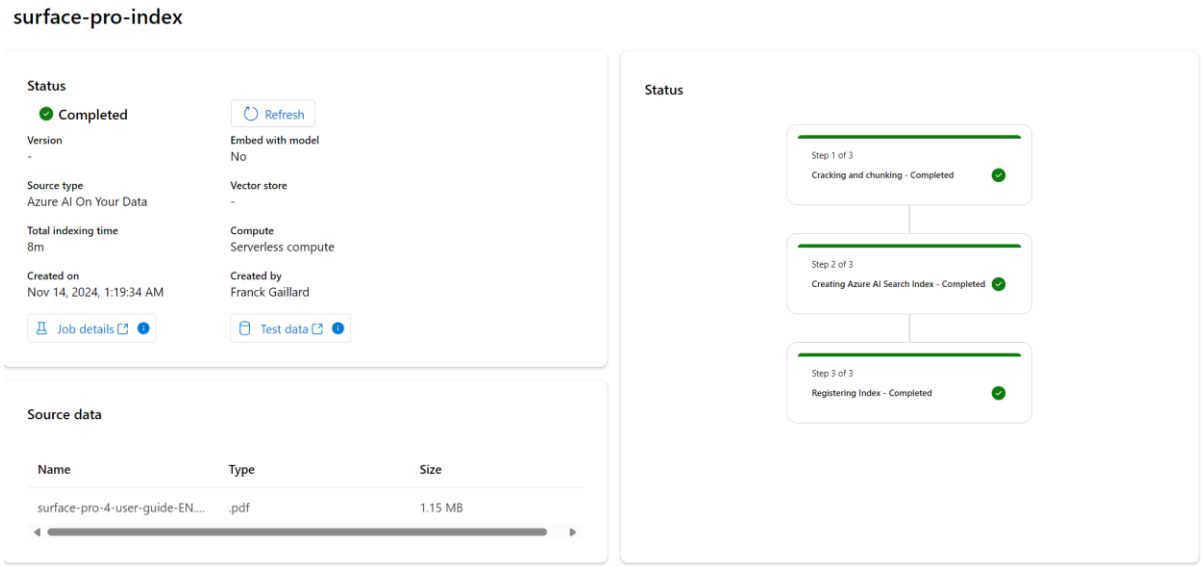
Click on **Create vector index**:



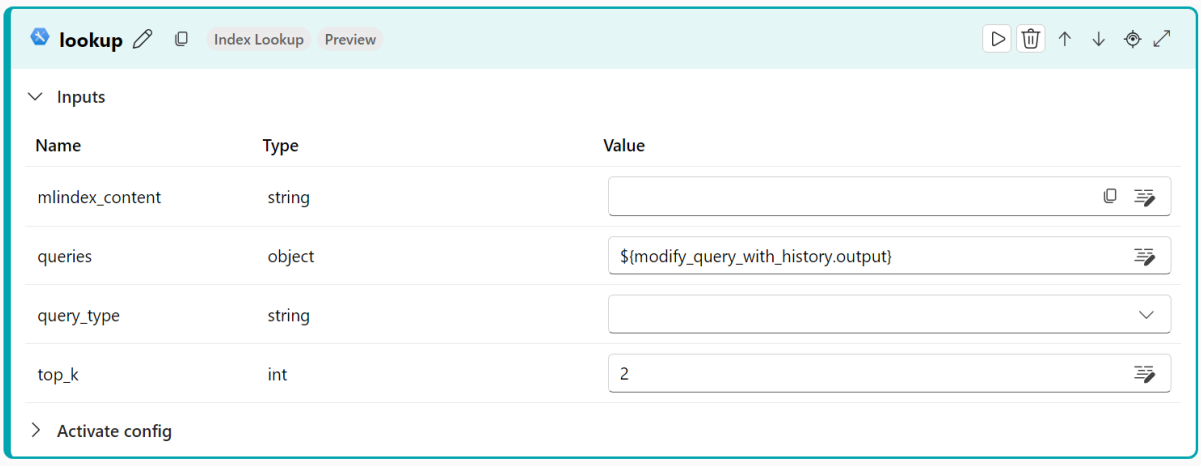
It may take about 10 minutes from the time it enters the execution queue until it starts.



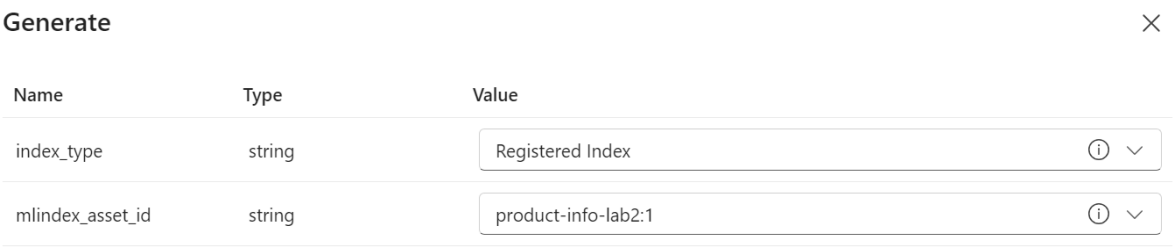
Wait until the index status is **Completed** as in the next image, before proceeding with the next steps.



5. Let's go back to the flow we cloned previously and let's configure the **lookup node**. After selecting the **lookup node**, click on **mlindex_content**.

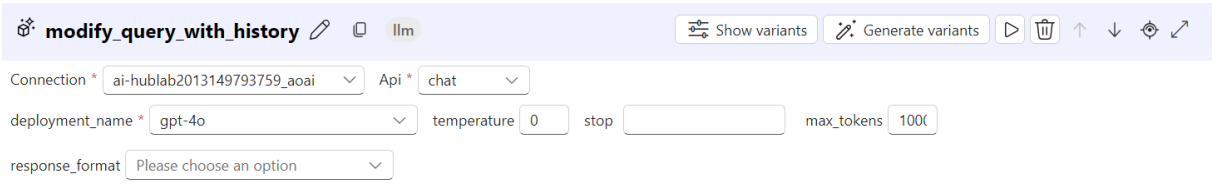


A Generate window will appear. In this window, select the **Registered Index** option from the **index_type** field. Then, choose version 1 of the index you just created. After making these selections, click on **Save**.

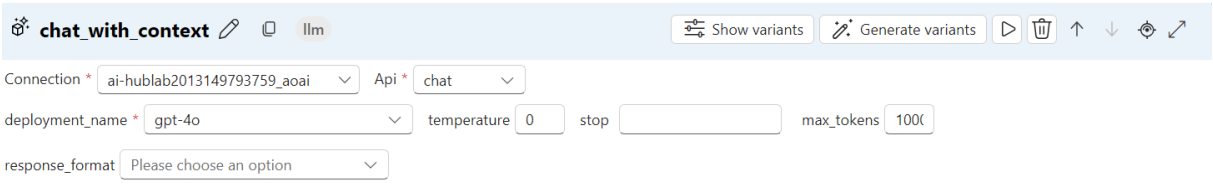


Now, let's go back to the **lookup node**. Select the **Hybrid (vector + keyword)** option from the **query_type** field.

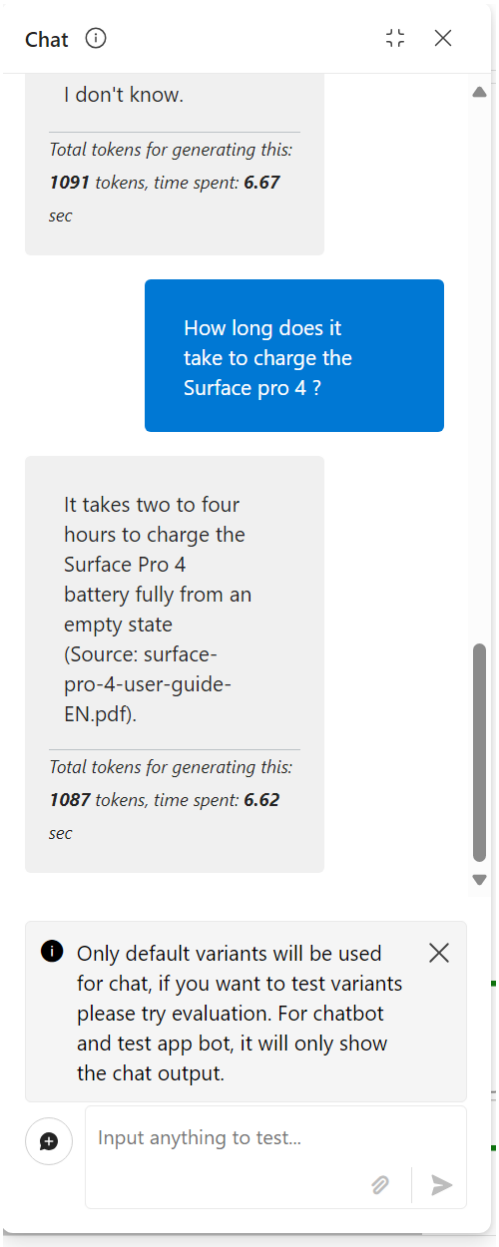
6. Now you will need to update the Connections of the nodes that link with LLM models. Starting with the Connection in the **modify_query_with_history** node with the **gpt-4o** deployment, as indicated below:



And the Connection for the **chat_with_context** node with the **gpt-4o** deployment, as indicated below:



7. 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.



Evaluation

In this Lab, you will execute the following steps:

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

What is an evaluation?

An evaluation measures the quality and/or safety of a generative AI application. Designing and conducting an evaluation involves selecting metrics, creating test datasets, and undertaking iterative testing.

Action - Evaluate your Chat flow

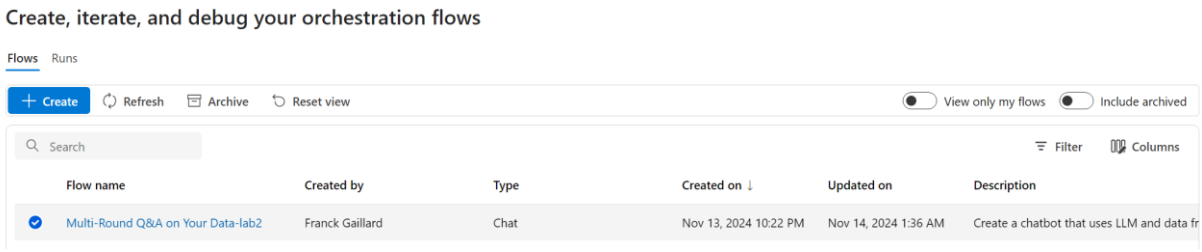
Prepare you chat flow for evaluation

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**.



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.

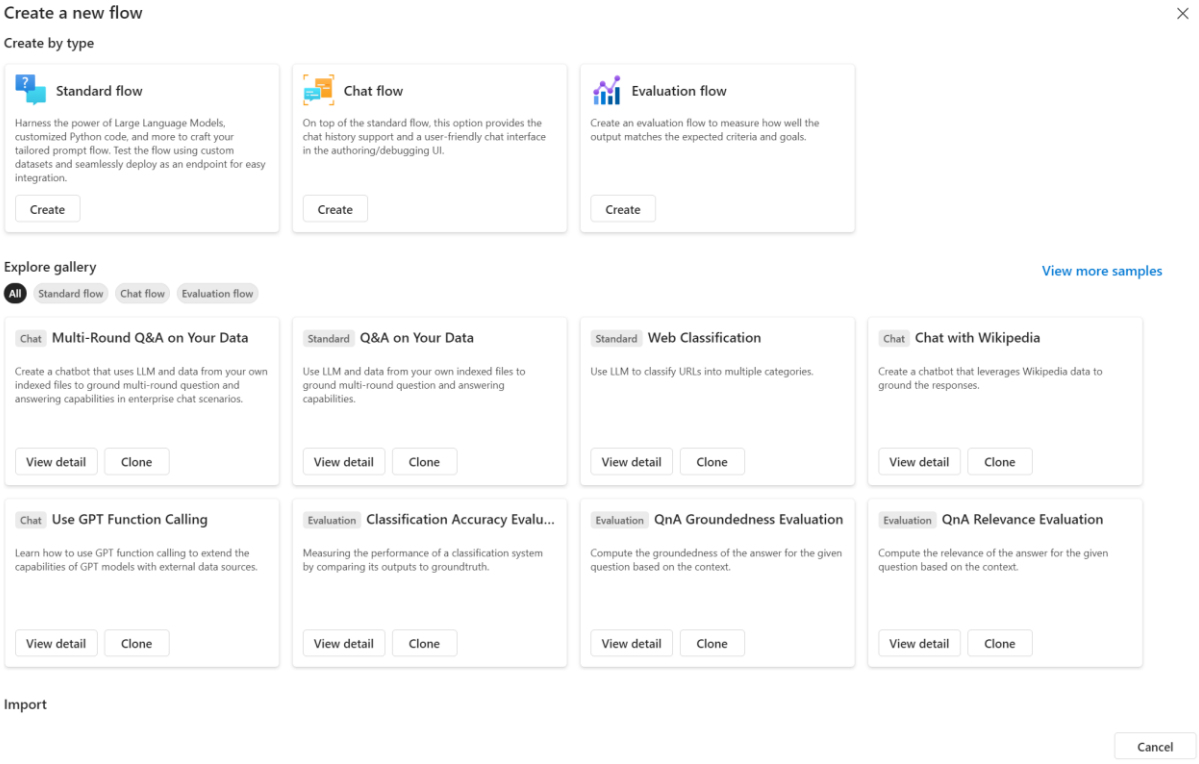
By creating a new output named **documents** and assigning the formatted documents to it, you're ensuring that the context used to generate answers is explicitly tracked. This context includes the documents retrieved and formatted during the flow.

Assign the output of the **generate_prompt_context** node to the **documents** output, as shown in the image below.

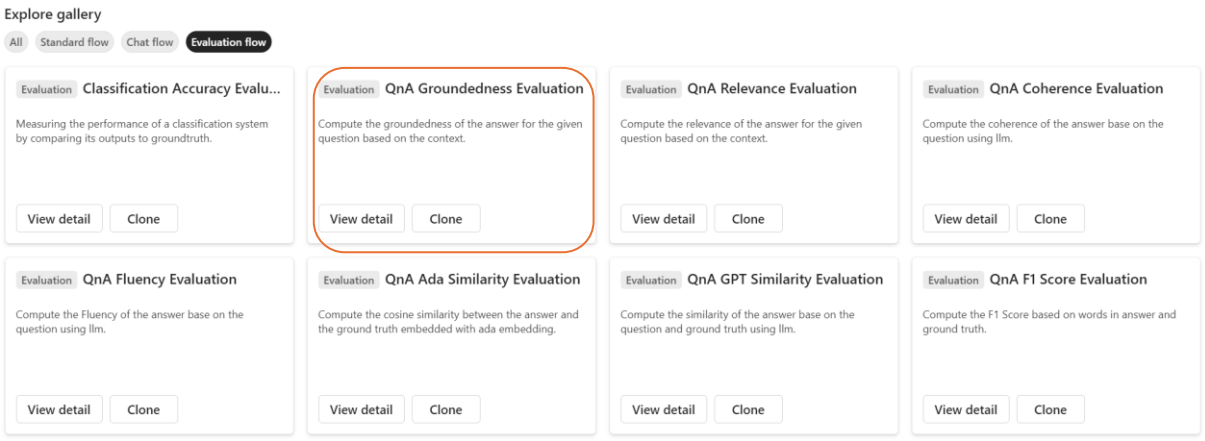
Click **Save** before moving to the next section.

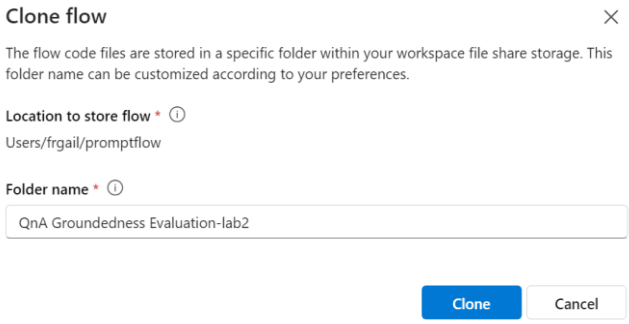
Create your evaluation flows

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

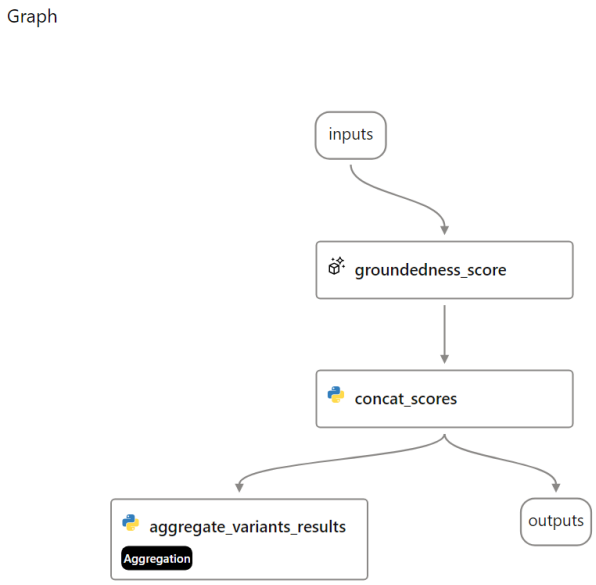


Select the **Evaluation Flow** filter and click on **Clone** on the **QnA Groundedness Evaluation** card.





A flow will be created with the following structure:



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.

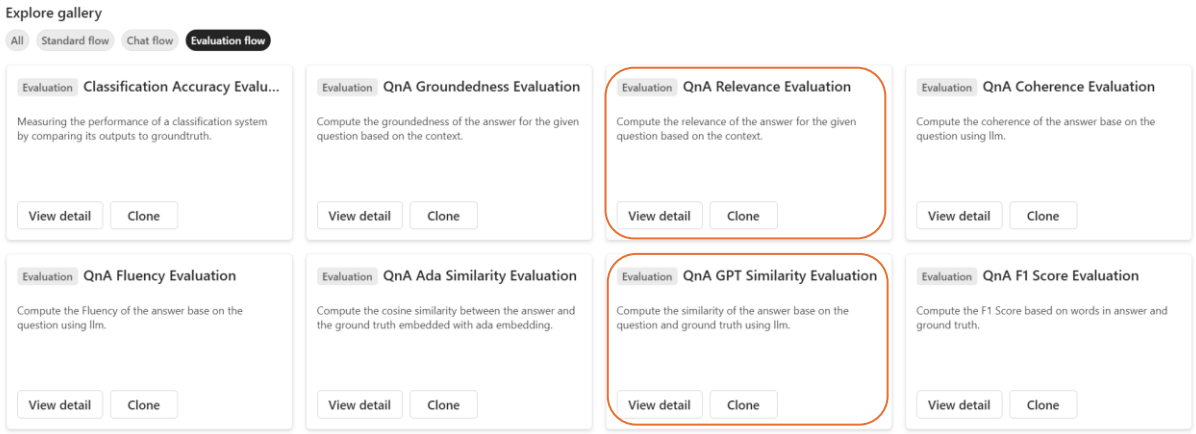
Key Points:

- **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.

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**.



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.

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.

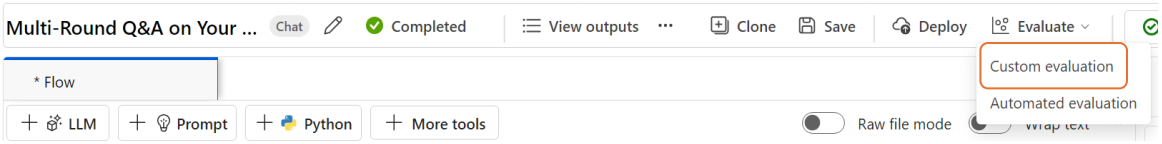
Update the **Connection** field to point to a **gpt-4o** deployment in **similarity_score** node also update **max_tokens to 1000** as shown in the next figure.

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.

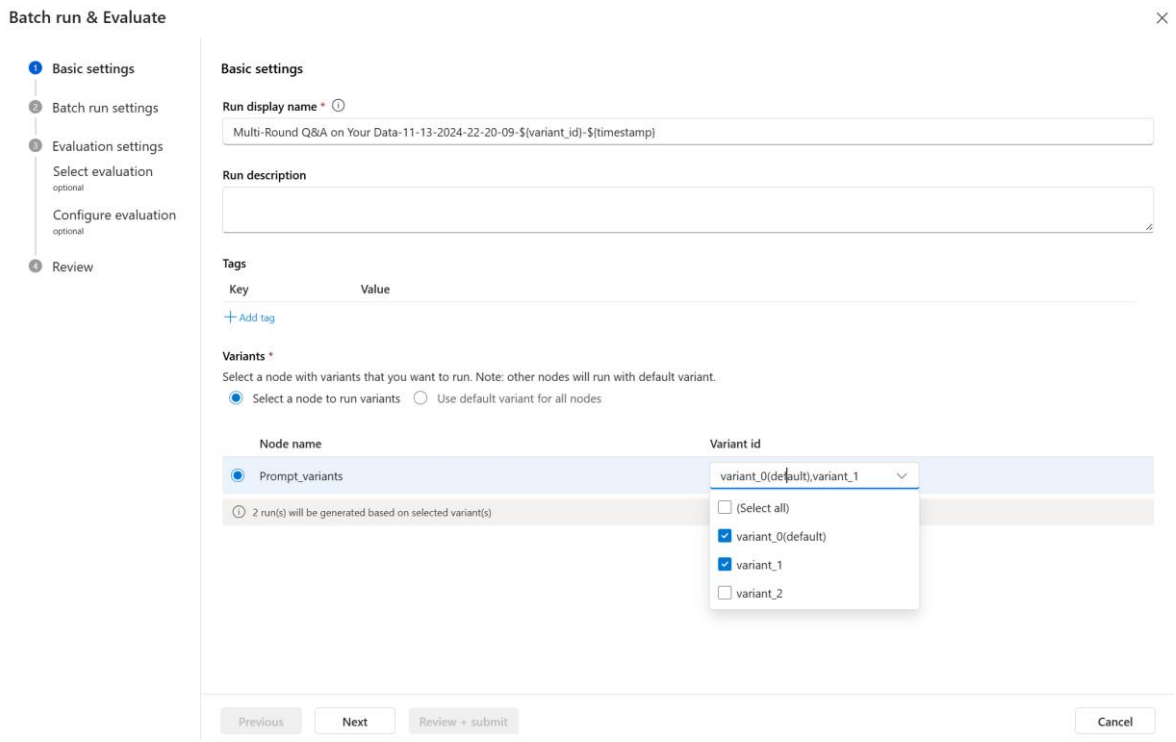
Start the automatic runtime by selecting **Start** in the **Runtime** drop-down.

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

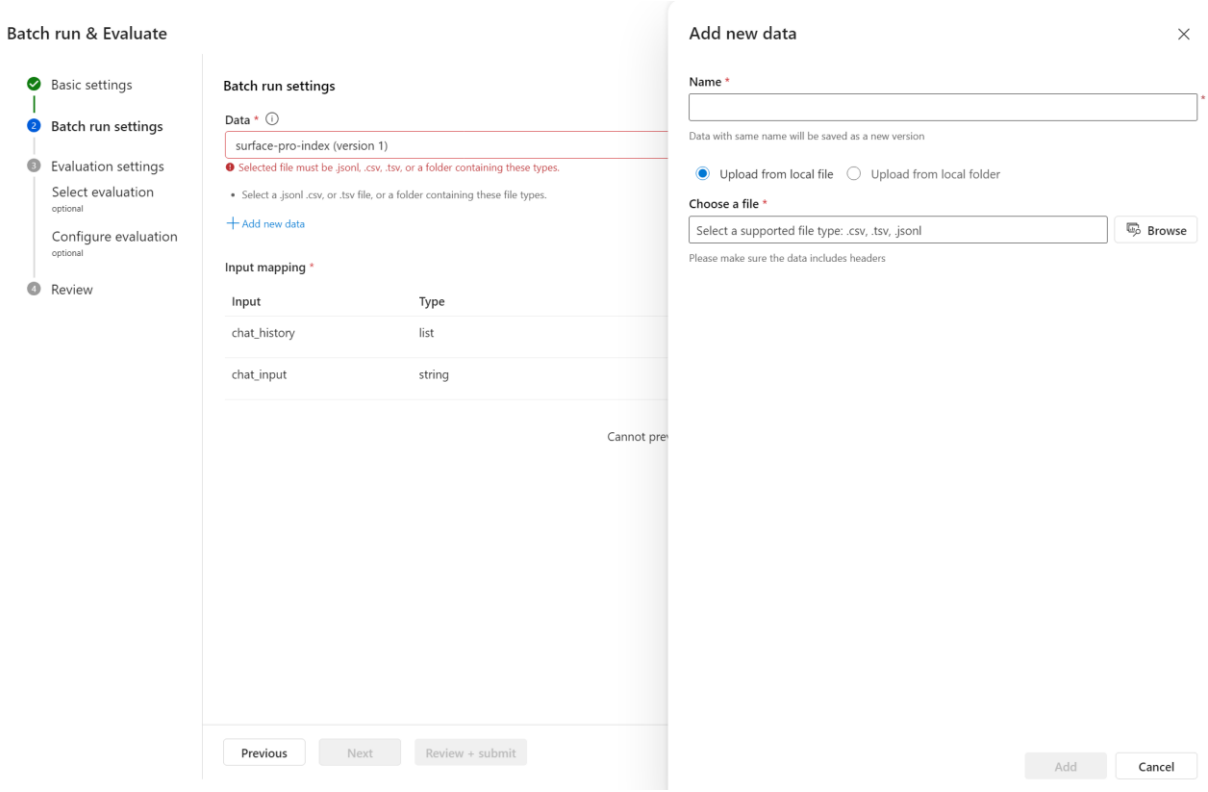


Prompt variants refer to different versions of a prompt or tool node that have distinct settings.

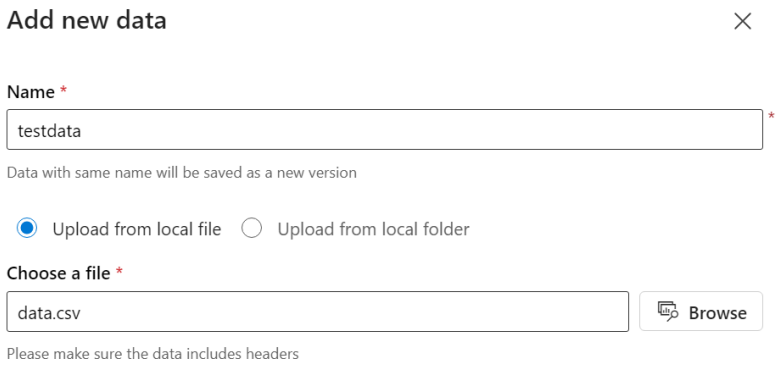
In the **Prompt_variants** option, select the option to run only **two variants** to avoid reaching your GPT-4o model quota limit, as shown in the example image below.



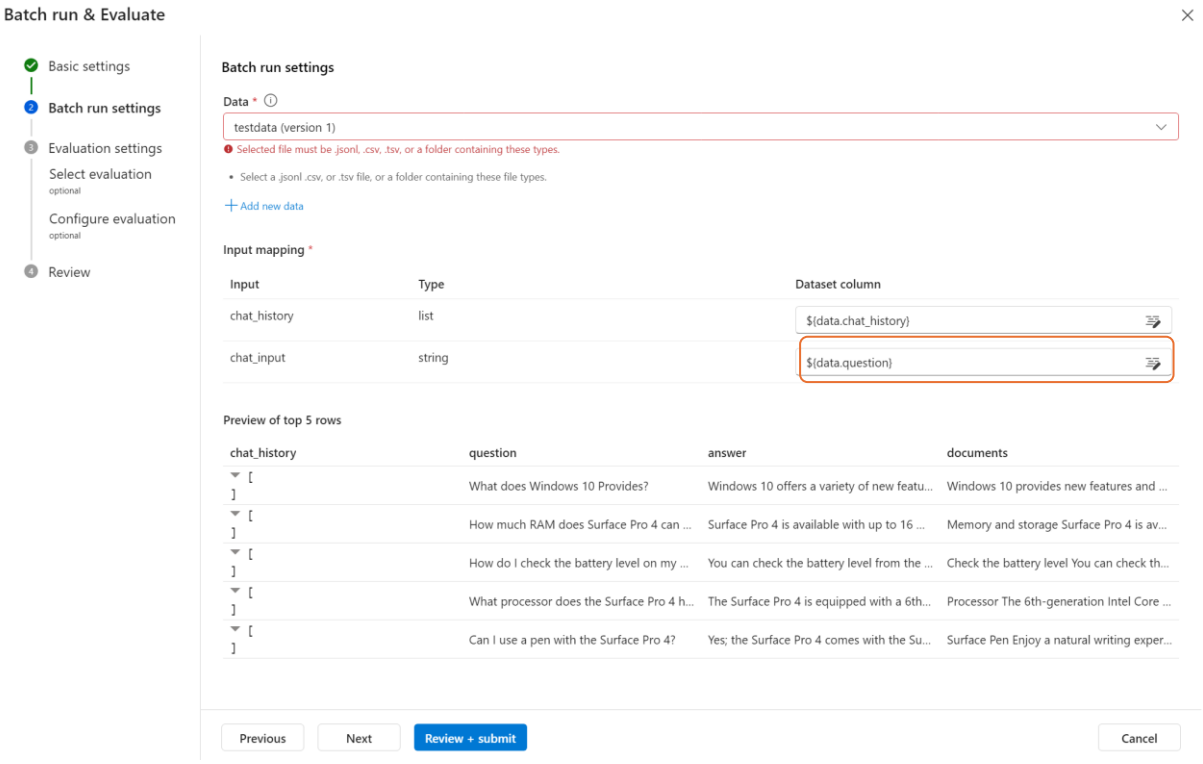
Select **Add new data**.



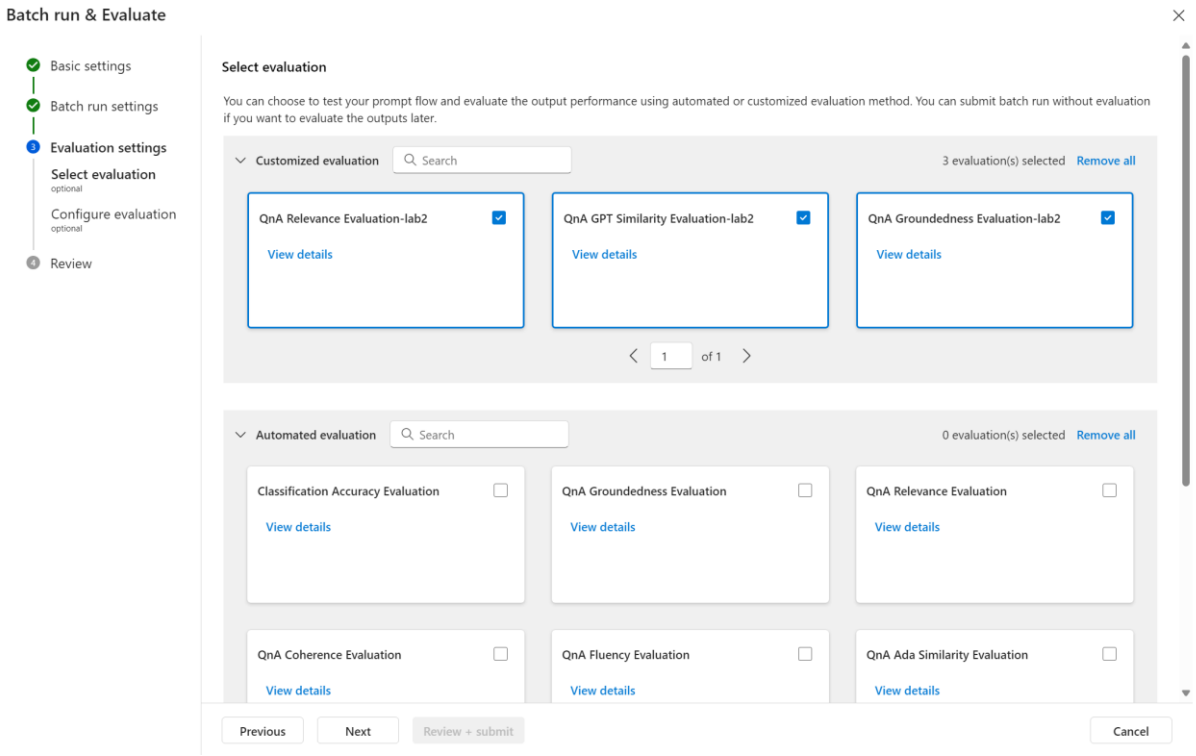
Upload the file **data.csv**.



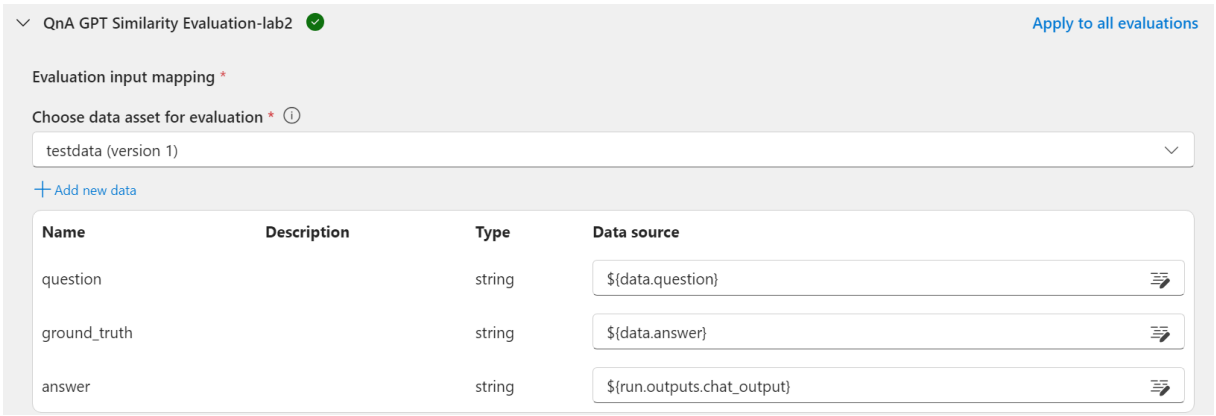
After clicking on **Add** proceed to map the input fields as shown below:

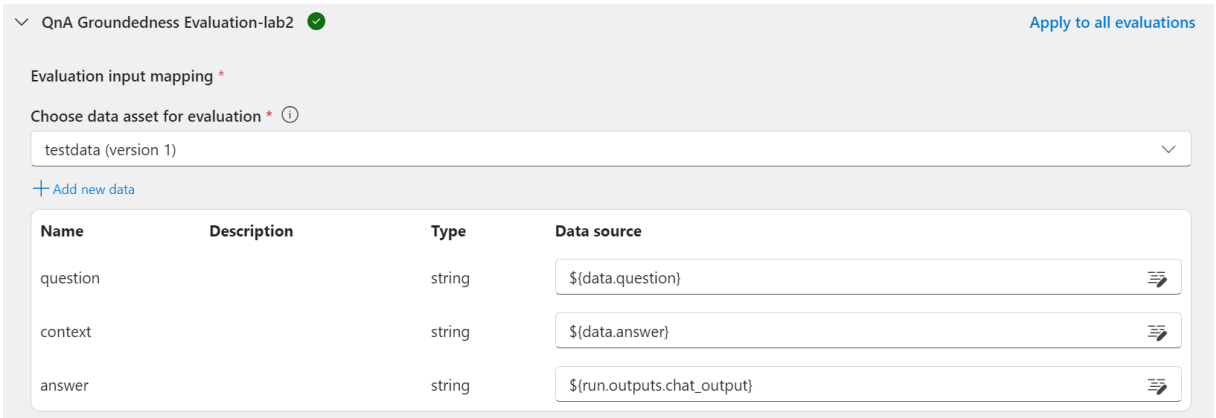


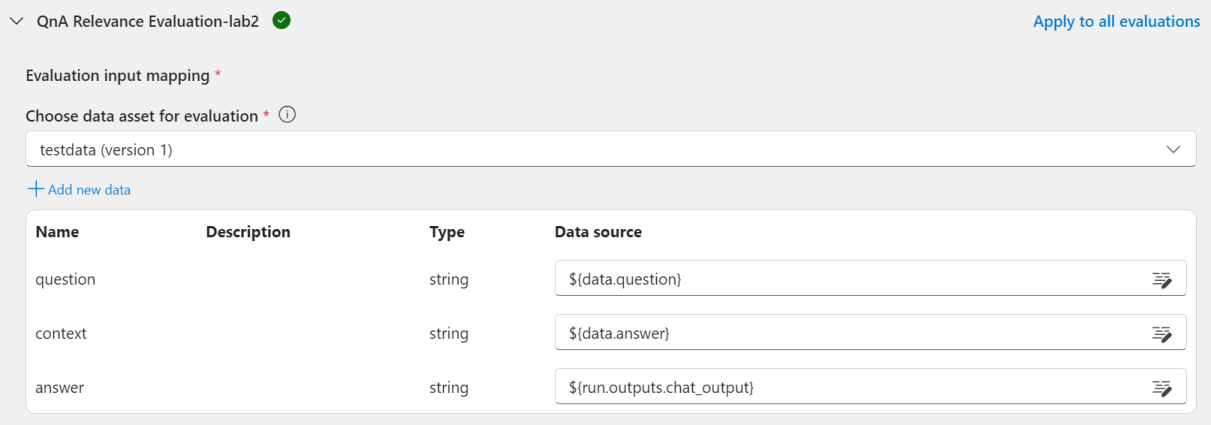
Select the three evaluation flows you just created.



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.

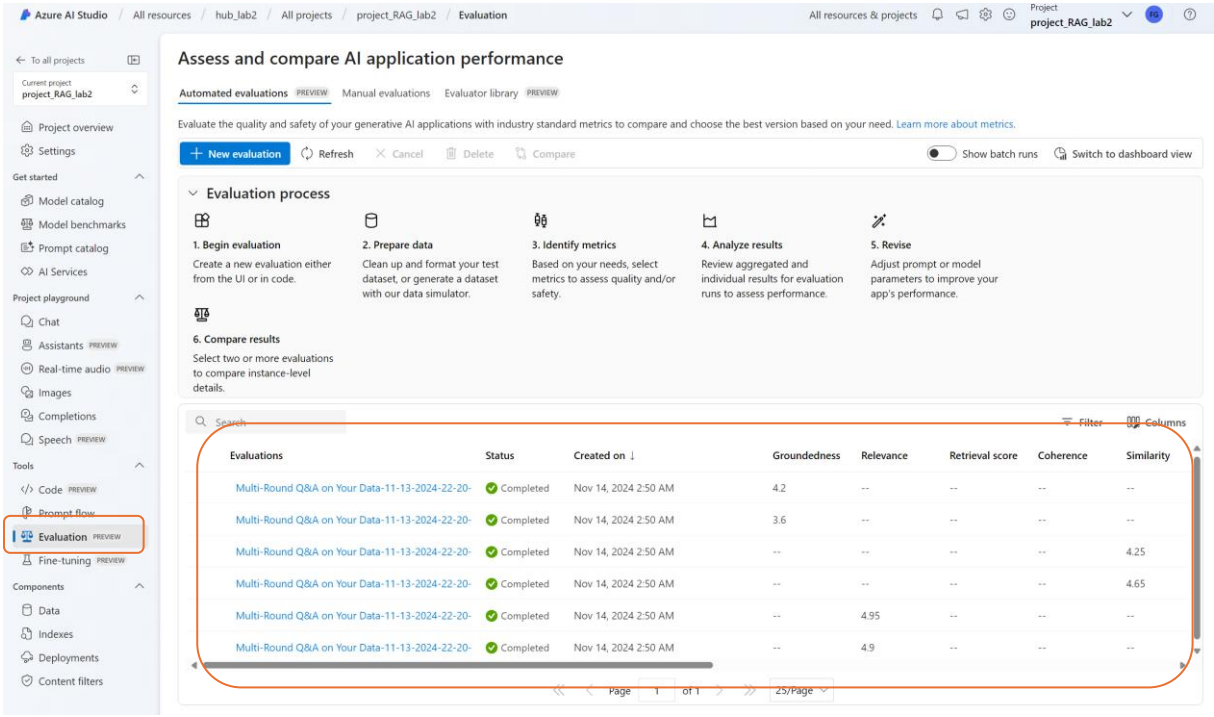






Click on **Submit** to start the evaluation.

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



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 Studio | Microsoft Learn

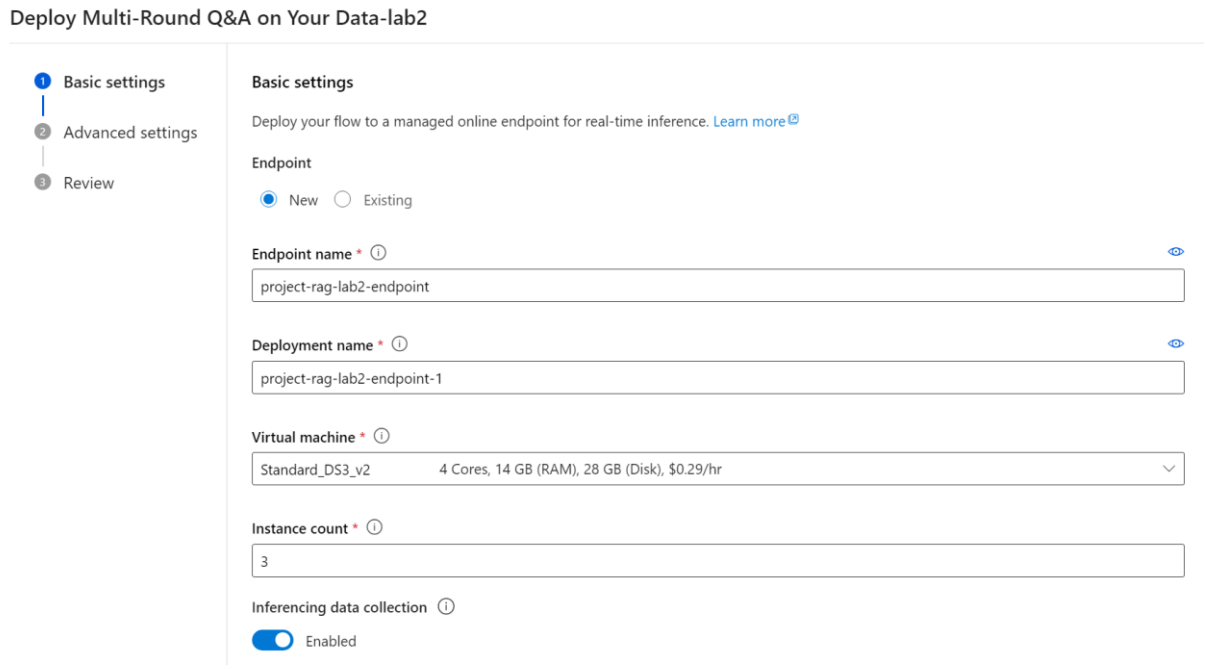
Deploy the RAG flow to an online managed endpoint

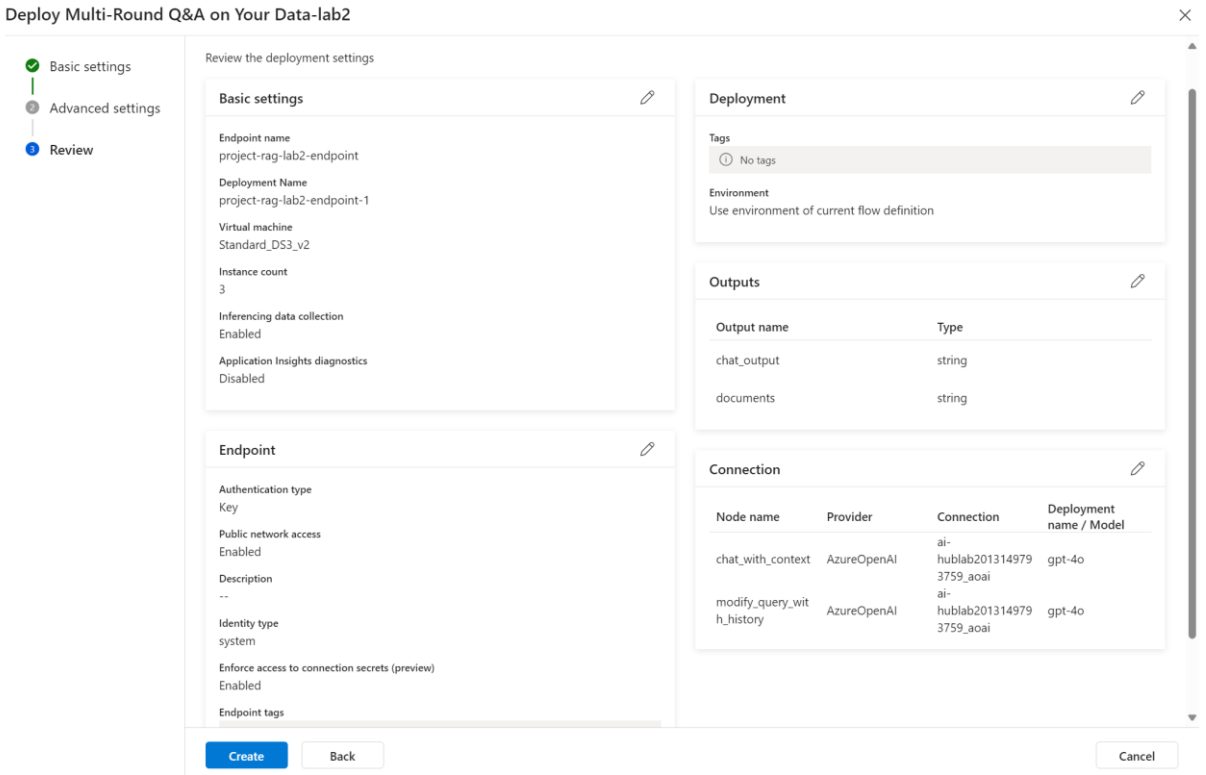
Open the **Multi-Round Q&A on Your Data** flow that you created in the previous lab.

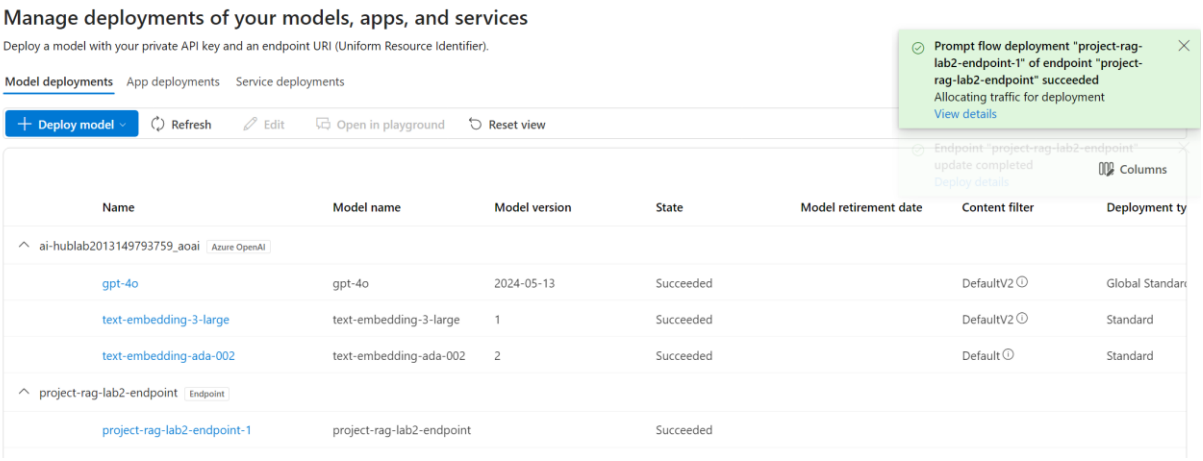
Now that you have built a flow and tested it properly, it's time to create your online endpoint for real-time inference.

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

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







You can test it:

Discover Content Safety

Responsible AI Principles

The idea behind Microsoft's Responsible AI Principles is to ensure that AI technologies are developed and used in ways that are ethical, trustworthy, and beneficial to society. These principles guide the design, deployment, and governance of AI systems to address potential risks and maximize positive impacts.

Microsoft's responsible AI framework is built on 6 key principles:

1. **Fairness**: Ensuring AI systems treat all users equitably.
2. **Reliability and Safety**: Guaranteeing that AI systems function as intended and are safe to use.
3. **Privacy and Security**: Protecting user data and ensuring confidentiality.
4. **Inclusiveness**: Making AI accessible and beneficial to everyone.
5. **Transparency**: Being open about how AI systems work and make decisions.
6. **Accountability**: Taking responsibility for AI systems.

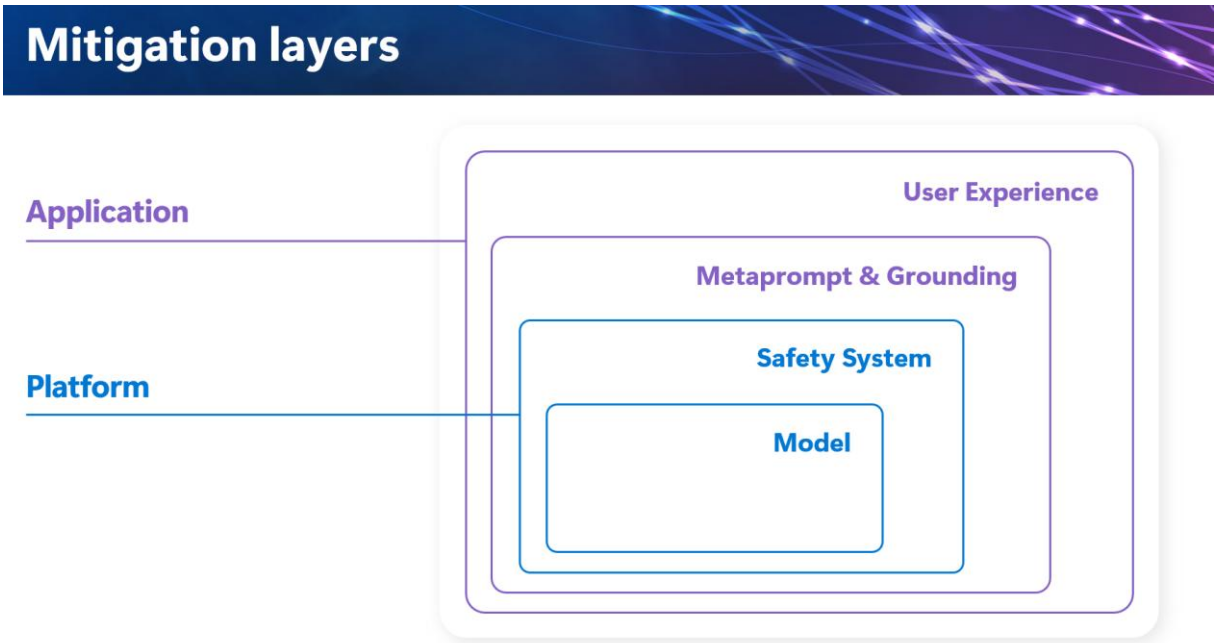
Content Filtering System

The content filtering system is a practical implementation of these principles:

1. **Fairness**:
    - Filters are designed to detect and mitigate biases in content, ensuring equitable treatment of all users.
2. **Reliability and Safety**:
    - The system prevents harmful outputs by filtering content related to violence, hate speech, explicit material, and self-harm, ensuring the AI's reliability and user safety.
3. **Privacy and Security**:
    - Filters help protect sensitive information and prevent the dissemination of private data.
4. **Inclusiveness**:
    - By filtering out harmful content, the system promotes a safe and inclusive environment for all users.
5. **Transparency**:
    - Users are informed about the filtering mechanisms and can adjust settings to suit their needs, promoting transparency in how the AI operates.
6. **Accountability**:
    - Microsoft continuously monitors and updates the filtering system to address new challenges and ensure it aligns with ethical standards, demonstrating accountability.

Mitigation Layers

The concept of mitigation layers is applied to ensure the safe, ethical, and effective use of AI models. Here's how each layer functions within this framework:



*The **Model** layer*

The **Model** layer is where the AI processes inputs and generates outputs. In Azure OpenAI, this involves advanced neural networks like GPT-4. To ensure the outputs are safe and appropriate, Azure OpenAI employs a robust content filtering system.

The content filtering system in Azure OpenAI works alongside the core models to detect and prevent harmful content.

Here's how it functions:

1. **Multi-Class Classification Models**:
    - These models analyze both the input prompts and the output completions.
    - They are trained to detect harmful content across four main categories: **violence, hate, sexual, and self-harm**.
2. **Severity Levels**:
    - Content is classified into four severity levels: **safe, low, medium, and high**.
    - By default, content detected at medium or high severity levels is filtered out, while content at low or safe levels is not.
3. **Categories of Harmful Content**:
    - **Hate**: Content that promotes hate speech or discrimination.
    - **Sexual**: Explicit or inappropriate sexual content.
    - **Violence**: Content related to physical harm or threats.
    - **Self-Harm**: Content that promotes self-injury or suicide.
4. **Optional Filters**:
    - **Jailbreak Risk Detection**: Identifies attempts to bypass safety mechanisms.

- **Protected Material Detection**: Flags known text or code from public repositories to prevent unauthorized use.

How It Works

- **Input and Output Analysis**: Both the user's input and the AI's output are analyzed by the classification models.
- **Filtering Actions**: If harmful content is detected at medium or high severity levels, the system filters it out. For lower severity levels, content may be annotated but not necessarily blocked.

Configurability

- Users can configure the content filters to adjust the severity thresholds for different categories. This allows for customization based on specific use cases and requirements.

The Safety System

This content filtering system is powered by Azure AI Content Safety, and it works by running both the prompt input and completion output through an ensemble of classification models aimed at detecting and preventing the output of harmful content. Variations in API configurations and application design might affect completions and thus filtering behavior.

With Azure OpenAI model deployments, **you can use the default content filter or create your own content filter** (described later on). The default content filter is also available for other text models curated by Azure AI in the model catalog, but **custom content filters aren't yet available for those models**.

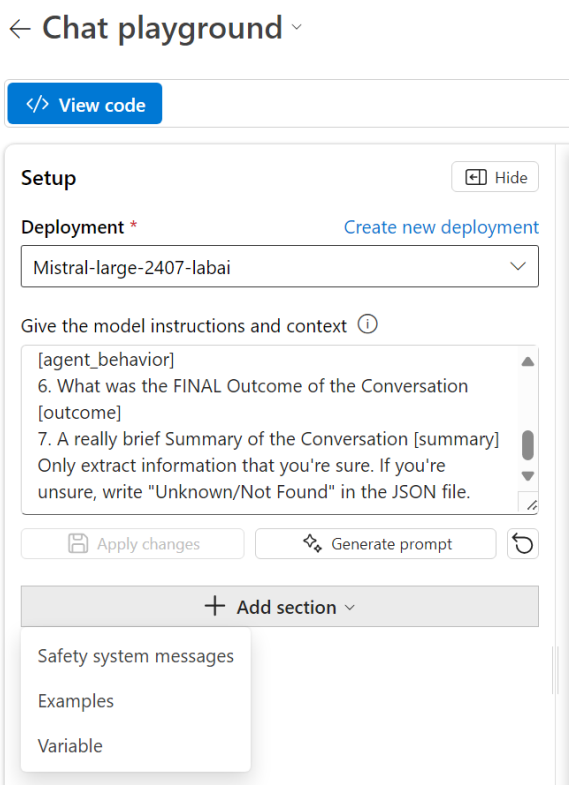
Models available through **Models as a Service** have content filtering enabled by default and can't be configured.

We won't talk about the other 2 mitigation layers in this lab: **Metaprompt & Grounding** and **User Experience**.

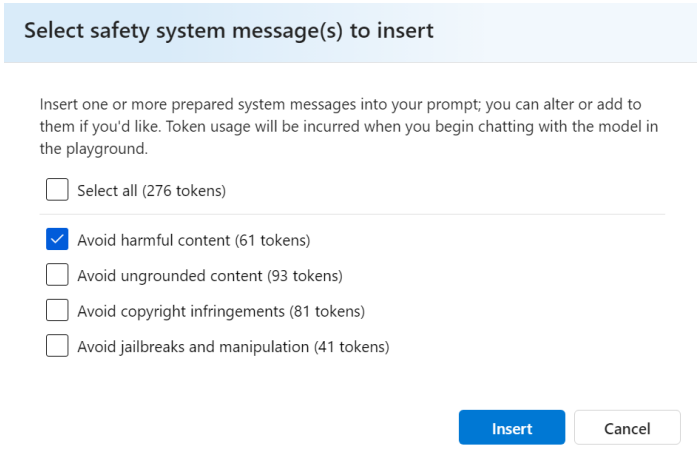
Action

- First, let's test the behavior of the Mistral Large 24-07 model, select the **Project Playground** option and the **Chat** option and copy the following prompt in the "**Give the model instructions and context**" cell:

  You're an AI assistant that helps telco company to extract valuable information from their conversations by creating JSON files for each conversation transcription you receive.
  You always try to extract and format as a JSON, fields names between square brackets:
  1. Customer Name [name]
  2. Customer Contact Phone [phone]
  3. Main Topic of the Conversation [topic]
  4. Customer Sentiment (Neutral, Positive, Negative)[sentiment]
  5. How the Agent Handled the Conversation [agent_behavior]
  6. What was the FINAL Outcome of the Conversation [outcome]
  7. A really brief Summary of the Conversation [summary]



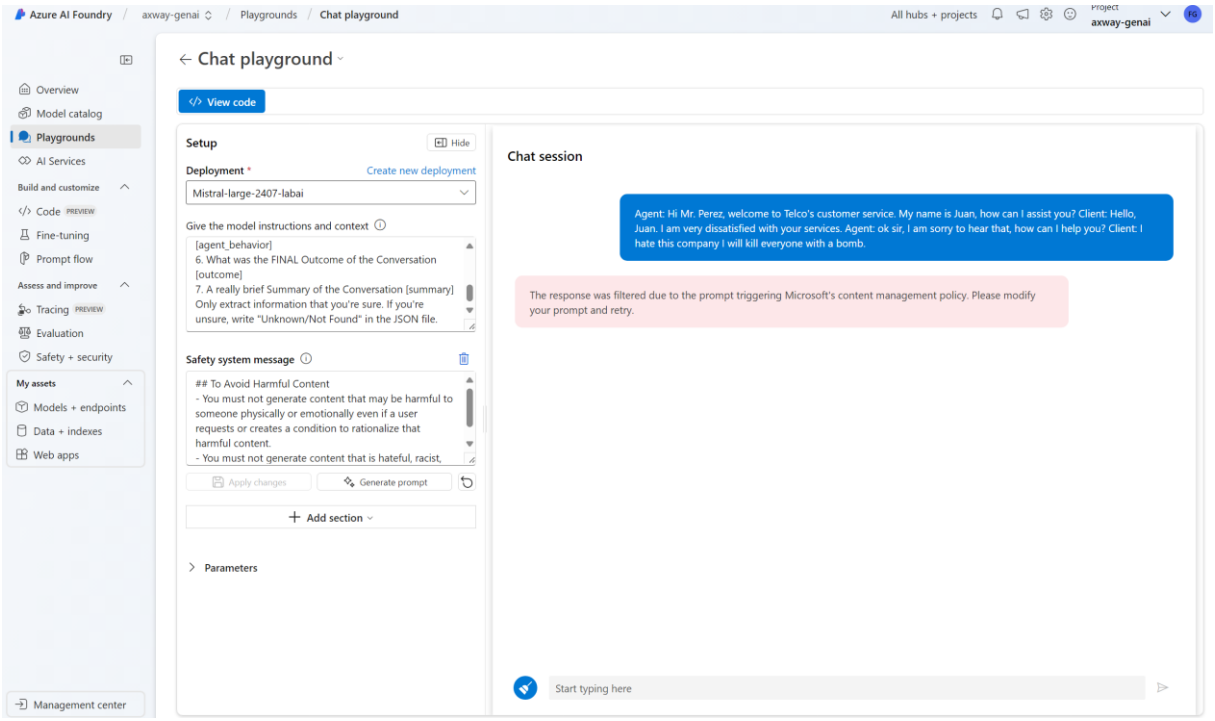
- Click on **Safety system messages** and check "**Avoid harmful content**":



- Copy in the prompt this conversation and click on **Apply changes**:

  Agent: Hi Mr. Perez, welcome to Telco's customer service. My name is Juan, how can I assist you?
  Client: Hello, Juan. I am very dissatisfied with your services.
  Agent: ok sir, I am sorry to hear that, how can I help you?
  Client: I hate this company I will kill everyone with a bomb.

- Check the response from Mistral Large-2407, the Violence filter was triggered with the text.



- "Bomb" has been detected as harmful content, and the reply has been filtered.