

# Introduction

**Data:** Please refer to the data folder in the GitHub repository shared with you during the event. Please refer to the Readme file in the GitHub repository for the data description and the problem statement

**Overview:** We'll build a chatbot powered by Azure AI services such as Azure AI Search, Azure Open AI, and Azure AI Studio. This chatbot will be capable of ingesting any document, responding to questions based on only the information available in the documents, and handling multi-turn conversations. Assuming this is our first AI chatbot project, we will detail the steps. We'll use Azure AI Studio to develop this solution and deploy a sample interactive web application.

If time permits, we'll also investigate how to change prompts and parameters like top\_k and their impact on responses. We will also try to give you a glimpse at other developer features, such as prompt flow and evaluation flow.

**Note:** In this step-by-step guide, we have used the resource configuration in table-1. You will receive your temporary values at the venue for this hackathon including a username.

Please follow the naming convention in Table 2 to create the names of the remaining Azure resources.

| S No | Configuration      | Name used in the documentation | Description                        |
|------|--------------------|--------------------------------|------------------------------------|
| 1    | User name          | Indonesia01                    | Azure User name provided           |
| 2    | Azure subscription | Tiger analytics-01             | Name of Azure Subscription         |
| 3    | Location           | Australia East                 | Location to create Azure Resources |
| 4    | Azure OpenAI       | openai-tigeranalytics-101      | Name of Azure OpenAI service       |

Table - 1

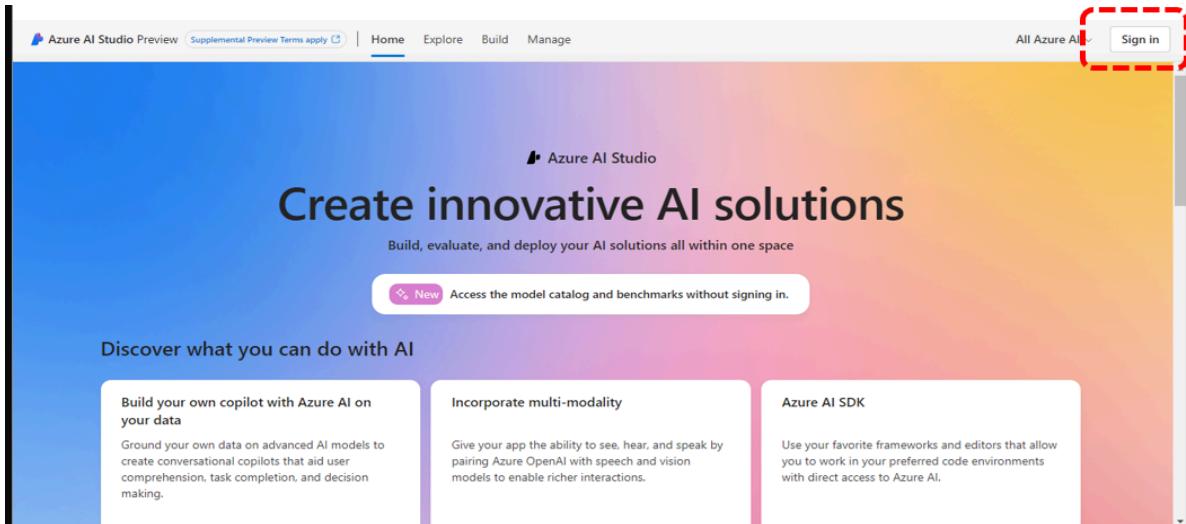
| S No | Configuration           | Naming Convention                   | Name used in documentation   |
|------|-------------------------|-------------------------------------|------------------------------|
| 1    | Resource group name     | rg-<User name>ai                    | rg-Indonesia01ai             |
| 2    | Azure AI search         | <User name>aisearch                 | Indonesia01aisearch          |
| 3    | Connection storage name | <User name>docs                     | Indonesia01docs              |
| 4    | Index name              | ai-hack-<User name>-index-<version> | ai-hack-indonesia01-index-v8 |
| 5    | Web app name            | <User name>-web app                 | Indonesia01-web app          |

Table 2

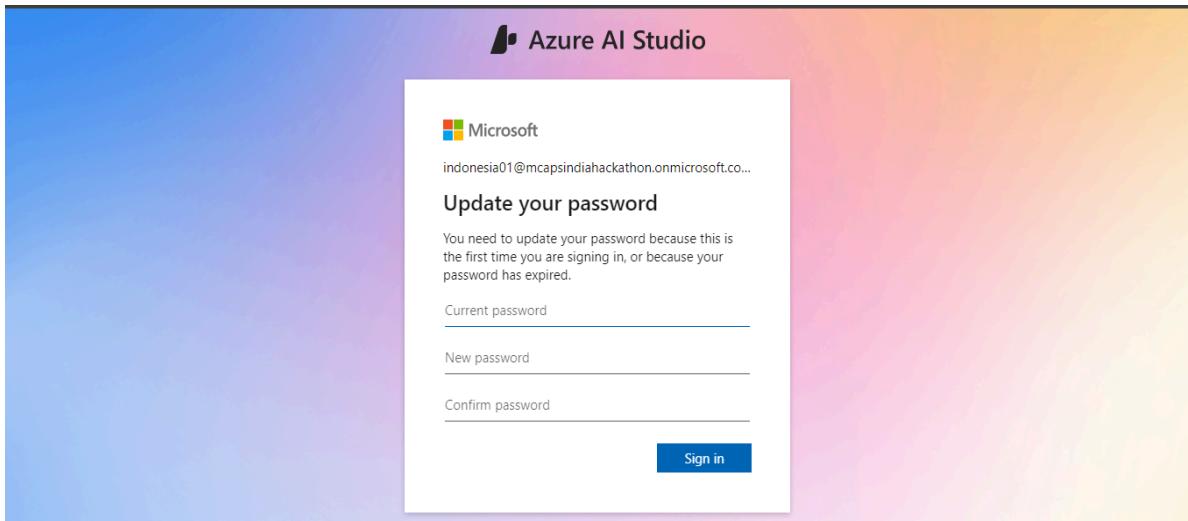
# Creating your project in AI Studio

## 1. Log in to the Azure AI Studio Portal

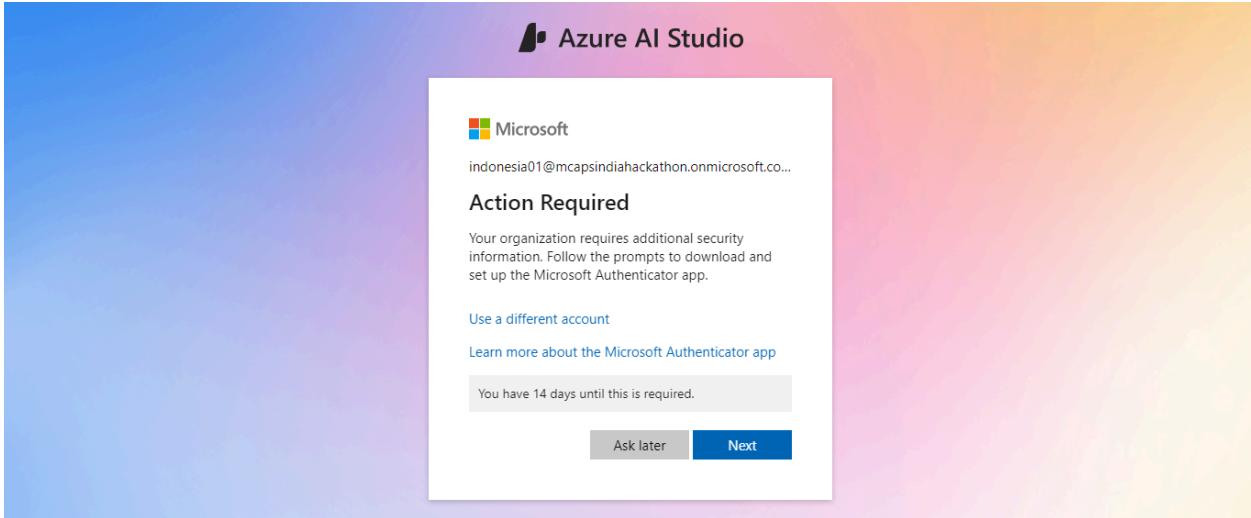
1. Please go to [Azure AI Studio](#) from your browser (please use Incognito/ private mode to avoid clashes with your existing work profile)
2. Click on the Sign-in Button and enter the credentials shared with you at the registration desk.  
(Please keep your slip with you till the end of the event)



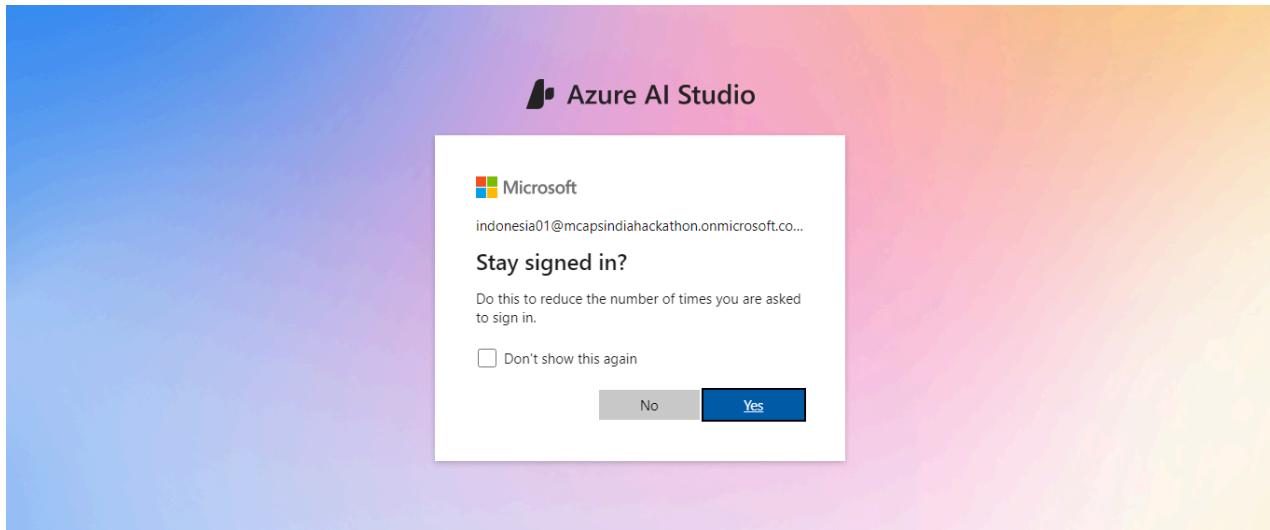
3. Please change the password after successful login.



4. Setup MFA/authenticator: Skip this step by clicking “Ask Later.” (If authenticator is a mandatory requirement, please choose “other method” for login and provide your phone number, you can sign-in through OTP)



5. Click on **Yes** in the next step.



## 2. Create a new project.

1. Go to the **Build** tab on the top and click on the **New AI Project** button

The screenshot shows the Azure AI Studio interface. On the left, there's a sidebar with links like 'Home', 'Get started', 'Model catalog', 'Model benchmarks', 'Prompt catalog', 'Azure OpenAI', and 'AI Services'. The main area displays 'Recent projects' with a table showing one entry: 'ai-hack-natarajansubr...' by 'natarajansubramai' in 'australiaeast' created on 'Jun 13, 2024 3:18 PM'. Below this is a section titled 'Jump to top tasks and tools'. A large modal window titled 'Create a project' is open. It has a 'Project details' tab selected. Inside, there's a 'Project name \*' input field containing 'ai-build-indonesia01-v1' which is highlighted with a red box. To the right of this input field is a 'Hub' dropdown menu with a 'Create a new hub' button also highlighted with a red box. Below the input fields, there's a note: 'There is no hub associated with your Azure subscription. You will be prompted to create a new hub on the next page.' and 'Projects using the same hub share security settings, and can share artifacts like data connections. [Learn more](#)'. At the bottom of the modal are two buttons: 'Create a project' (in blue) and 'Cancel'.

2. Add a project name with the following naming convention in the **Create Project** window.  
**“ai-hack-” + login ID + version number.** E.g. “ai-hack-indonesia01-v1”
  3. Click on **Create a new hub**. This should open a new window for the hub details.
4. In the new window,
- a) Under the **Hub name**, your user ID will be populated by default. If not, please add an appropriate resource name (userid+“ai”, Ex: “Indonesia01ai”). If your user ID has dots in it, remove it.
  - b) Under **Subscription**, select the default option. It will look like **Tiger analytics-<number>**, e.g. Tiger analytics-01
  - c) Under the **Resource group**, select “(new) + <the name you added in the Hub name>,” which will be displayed by default.
  - d) Under the **Location**, select **Australia East**.

**Create a project**

Project details

2 Create a hub

3 Review and finish

Create a hub for your projects  
A hub is the collaboration environment for your team to share your project work, model endpoints, compute, (data) connections, and security settings. [Learn more](#)

Do you need to customize security or the [dependent resources](#) of your hub? [Go to Azure Portal](#)

**Hub name \***  
Indonesia01\_ai

**Subscription \*** ⓘ [Create new subscription](#)  
Tiger analytics-01

**Resource group \*** [Create new resource group](#)  
(new) rg-Indoenasia01ai

**Location \***  
Australia East

[Help me choose](#)

**Connect Azure AI Services or Azure OpenAI \*** ⓘ [Create new AI Services](#)  
openai-tigeranalytics-101

**Connect Azure AI Search** [Create new AI Search](#)  
aihubsearch2401

[Back](#) [Next](#) [Create a project](#) [Cancel](#)

- e) Under **Azure OpenAI**, select the dropdown menu and select **the available Azure Open AI resource**. It will be in the format **openai-tigeranalytics-<number>**. e.g. **openai-tigeranalytics-01**
- f) Under **Connect Azure AI search**, select the option from the drop-down that looks like "**aihubsearch24<number>**" and click "**Next**"
- g) Review the details and Click on "**Create a project**" only if the resources are in the format as in the above screenshot. Otherwise, recheck the services, and if the service names do not match those shown below, please contact one of the event coordinators.
- h) Please wait while the Azure services are created for you. It might take a couple of minutes.

### Create a project

- ✓ Project details
- ✓ Create a hub
- Review and finish

**Review and finish**

The following resources will be created for you, along with required dependencies. The creation of the first hub and project may take a few minutes to complete. [Learn more about hubs and dependencies](#).

**Hub**  
Name: Indonesia01\_ai  
Subscription: Tiger analytics-01  
Resource group: rg-Indonesia01\_ai  
Location: australiaeast

**Project**  
Name: ai-build-Indonesia01-v1  
Subscription: Tiger analytics-01  
Resource group: rg-Indonesia01\_ai

**Linked Azure OpenAI**  
Name: openai-tigeranalytics-101

**AI Services**  
Name: ai-Indonesia01\_ai

**Linked Azure AI Search**  
Name: aihubsearch2401

Back
Create a project
Cancel

### Create a project

- ✓ Project details
- ✓ Create a hub
- Review and finish

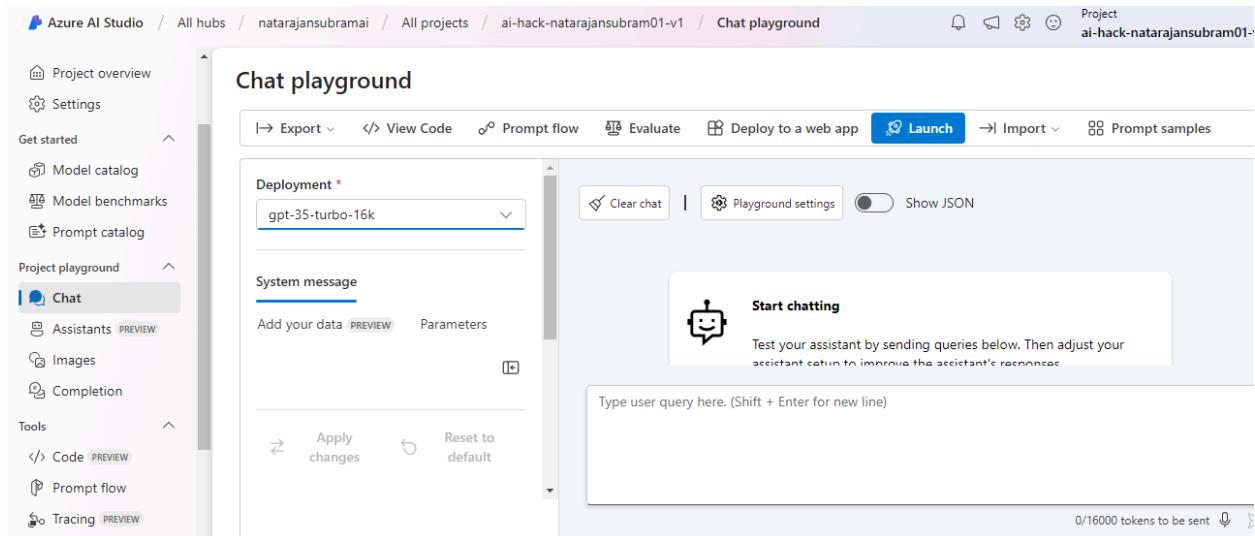
**Review and finish**

The following resources will be created for you, along with required dependencies. The creation of the first hub and project may take a few minutes to complete. [Learn more about hubs and dependencies](#).

| Resource                | Type   |
|-------------------------|--|
| ai-build-Indonesia01-v1 | AI project <span style="font-size: small;">(i)</span>      |
| Indonesia01_ai          | AI hub <span style="font-size: small;">(i)</span>          |
| ai-Indonesia01_ai       | AI Services <span style="font-size: small;">(i)</span>     |
| stindonesia0            | Storage account <span style="font-size: small;">(i)</span> |
| kv-indonesi             | Key vault <span style="font-size: small;">(i)</span>       |

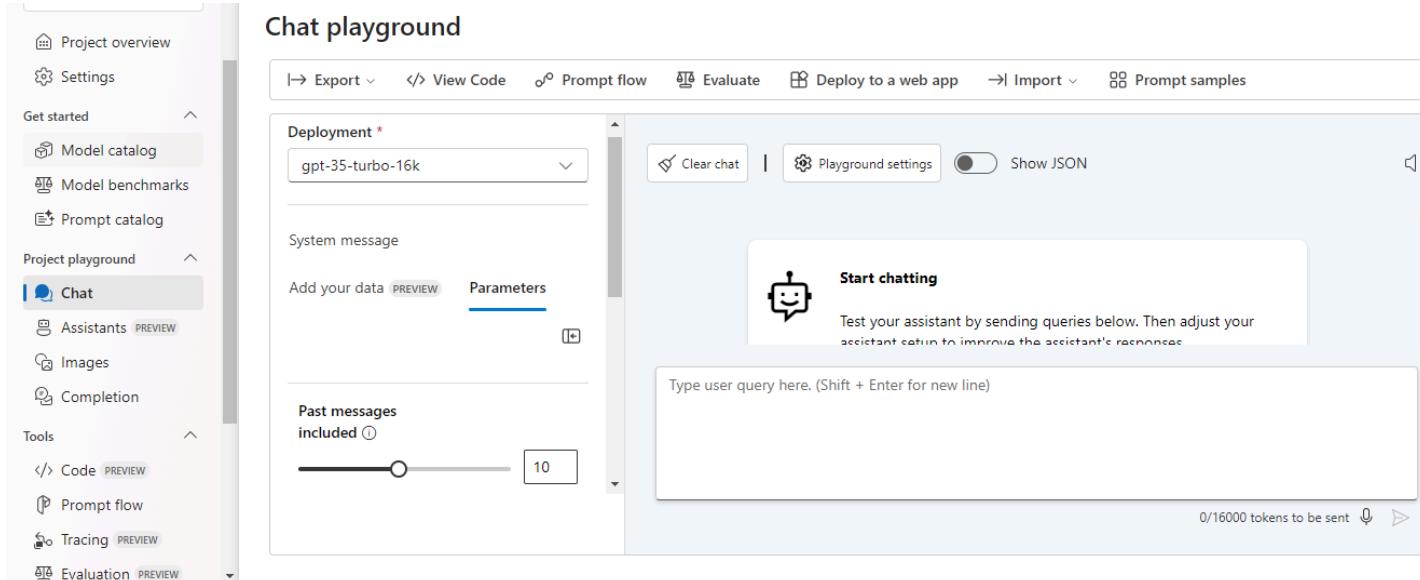
Validate resources...
Create a project
Cancel

- i) It will automatically redirect you to the **Project Overview** page. Click **Chat** under **Project Playground** to start with the next step.



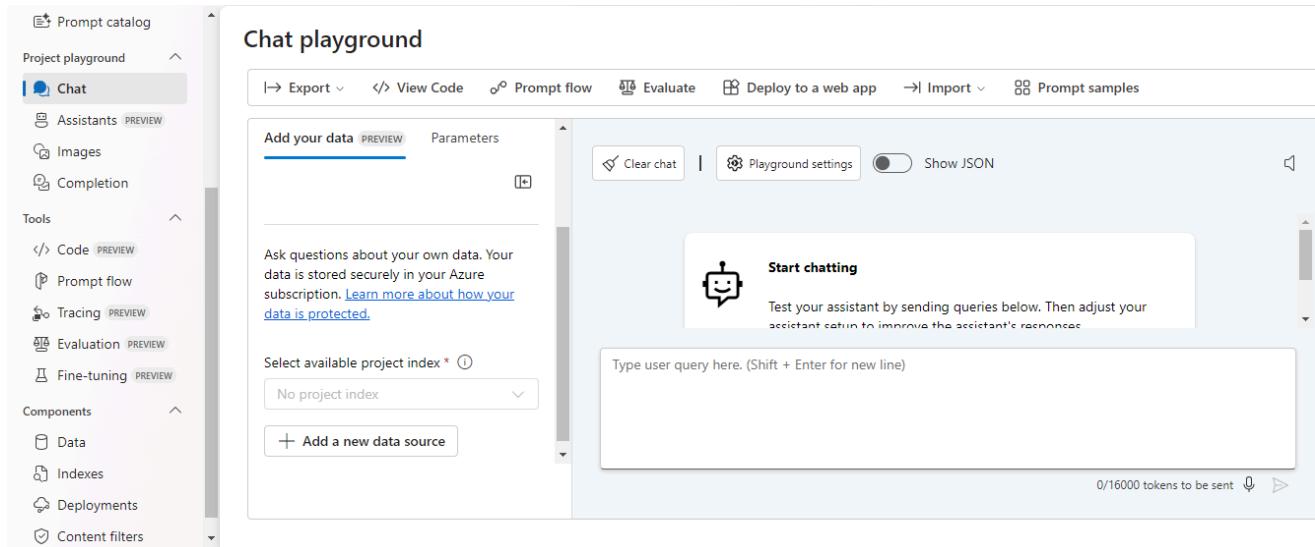
### 3. Building your project

1. On the project page, notice the left-hand side navigation sidebar. Click on the **Chat** under Project playground if you haven't yet. You will see the Playground in the screenshot below.

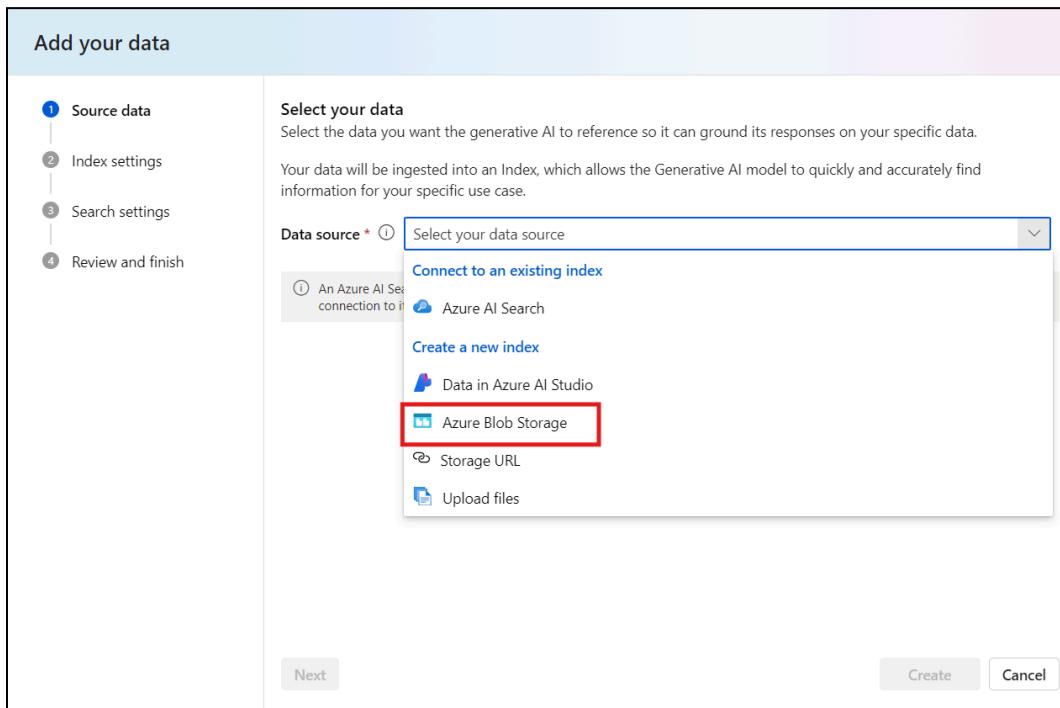


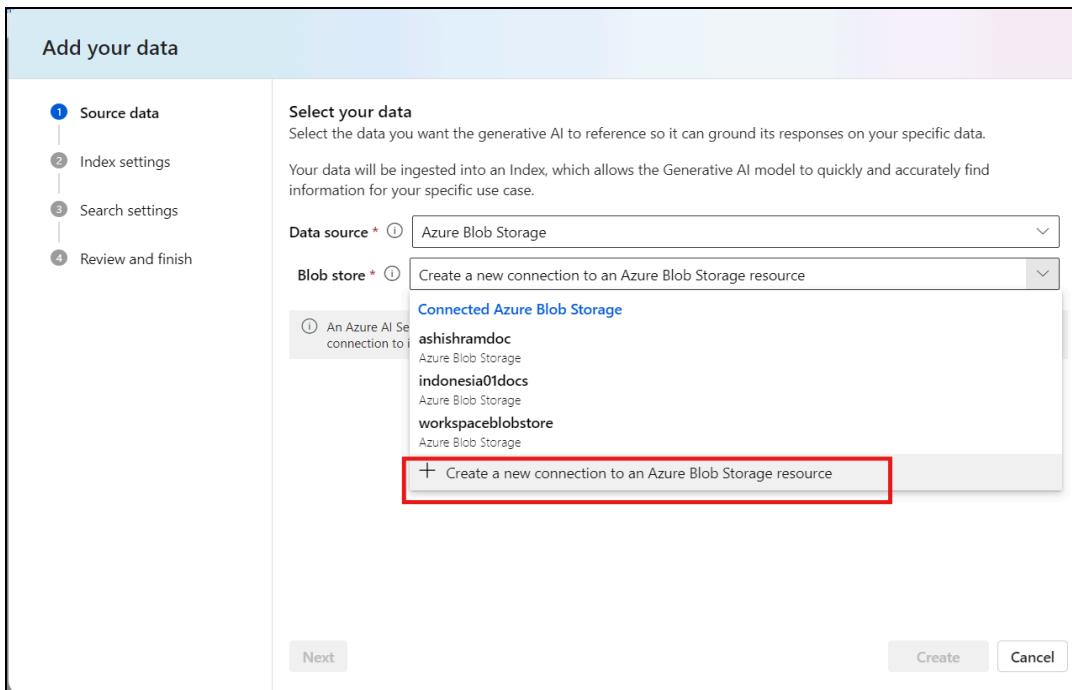
2. Let's confirm if we are connected to the Azure OpenAI Service. On the right-hand side panel of the project playground, select the model as **gpt-35-turbo-16k** under deployment dropdown. Then, go to the Parameters tab and change the **Past messages included** to **10**.

3. Now, let's add our data. On the left-hand side of the playground, go to **Add your Data** and click on **Add a new data source**.



4. Open the dropdown menu and in the Data source select "**Azure Blob Storage**" and under Blob Storage, select "**Create a new connection to Azure Blob Storage resource**". This will open a new window. Follow the instructions below to do that.





- i) In the popup window, the **subscription ID** will be populated by default. Select the **storage account**, choose **aihubsampledocs24<numericid>** and **blob container as docs**, and select the **Authentication method** as **Microsoft Entra-Based Id**. Enter the **connection name** appropriately, “**user-id + document**”. Ex: **“Indonesia01docs”** and click on **Create Connection**.

### Add a connection to external assets

Service \*  
Azure Blob Storage

Account selection method \*  
 Manually enter account information

Subscription ID \*  
Tiger analytics-01 (57a36344-3906-4293-9991-5010c5255d5e)

Storage account \*  
aihubsampledocs2401 (shared-resources)

Blob container \*  
docs

Authentication method \* ⓘ  
Microsoft Entra ID based

Connection name \* ⓘ  
Indonesia01docs

Access ⓘ  
Project

Create connection Cancel

- ii) Select data source as “Azure Blob Storage”. In the Blob store, select the connection you created earlier. E.g “**Indonesia01docs**” and click on **Create**.

**Add your data**

1 Source data  
2 Index settings  
3 Search settings  
4 Review and finish

**Select your data**  
Select the data you want the generative AI to reference so it can ground its responses on your specific data.  
Your data will be ingested into an Index, which allows the Generative AI model to quickly and accurately find information for your specific use case.

Data source \* ⓘ Azure Blob Storage

Blob store \* ⓘ Select a Blob Store

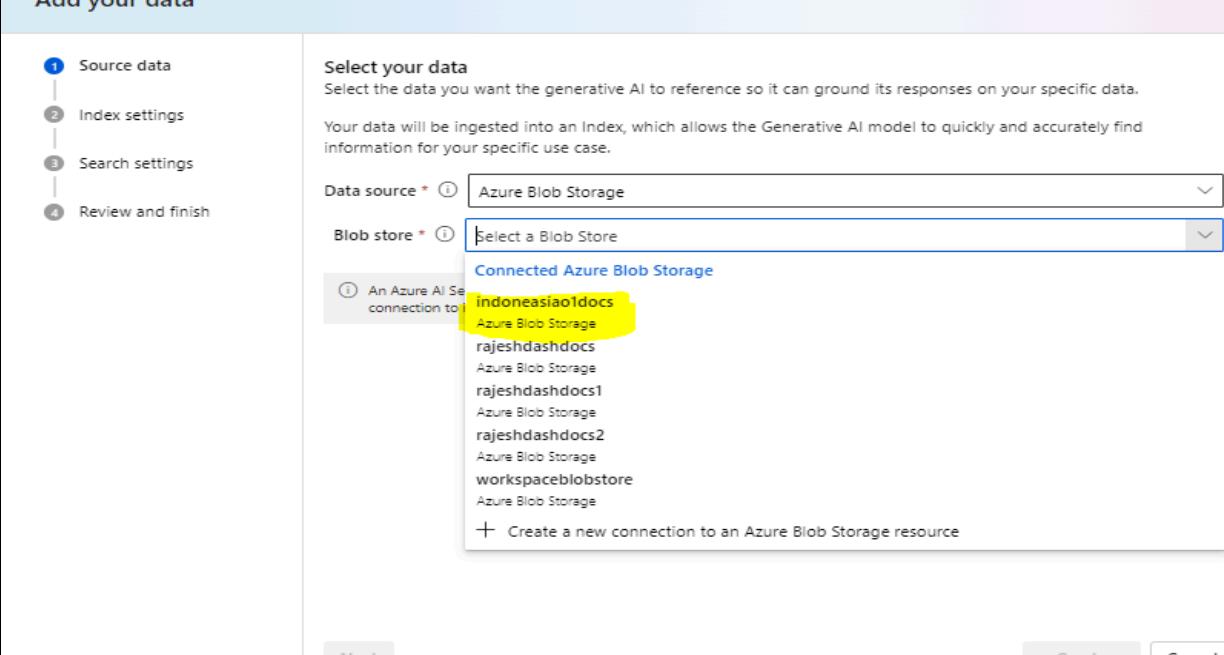
An Azure AI Search connection to: indoneasiao1docs

Connected Azure Blob Storage

- indoneasiao1docs
- Azure Blob Storage
- rajeshdashdocs
- Azure Blob Storage
- rajeshdashdocs1
- Azure Blob Storage
- rajeshdashdocs2
- Azure Blob Storage
- workspaceblobstore
- Azure Blob Storage

+ Create a new connection to an Azure Blob Storage resource

Next Create Cancel



- iii) Select the **data** folder, and click **next**. You can also click on the “**data**” folder to go inside it and view individual PDF files. You can select individual files or the whole folder for the next step.

**Add your data**

1 Source data  
2 Index settings  
3 Search settings  
4 Review and finish

**Select your data**  
Select the data you want the generative AI to reference so it can ground its responses on your specific data.  
Your data will be ingested into an Index, which allows the Generative AI model to quickly and accurately find information for your specific use case.

Data source \* ⓘ Azure Blob Storage

Blob store \* ⓘ indoneasiao1docs

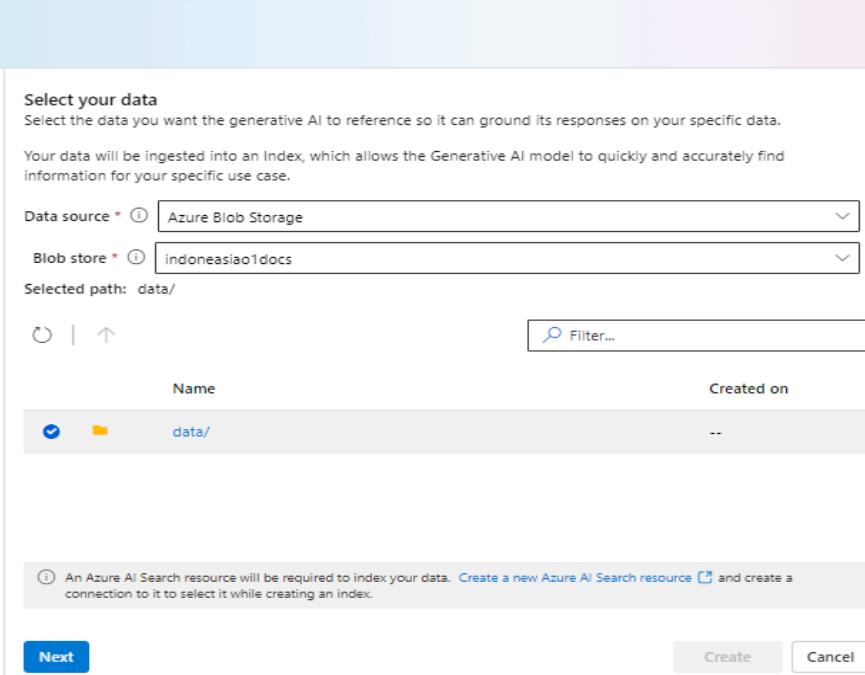
Selected path: data/

Filter...

| Name  | Created on |
|-------|------------|
| data/ | --         |

An Azure AI Search resource will be required to index your data. [Create a new Azure AI Search resource](#) and create a connection to it while creating an index.

Next Create Cancel



- iv) Select the default “Azure AI search” under Select **Azure AI Search** service. In the **Index Settings**, use the index name as follows: “**ai-hack-<userid>-index-<version>**”. Example **ai-hack-indonesia01-index-v1**

Add your data

Source data

Index settings

Search settings

Review and finish

**Index settings**  
Configure your index

**Index storage \***  
Azure AI Search

**Select Azure AI Search service \*** ⓘ  
AzureAISeach

[Create a new Azure AI Search resource](#) ⓘ

**Index name \*** ⓘ  
ai-build-indoensasia01-index-v1

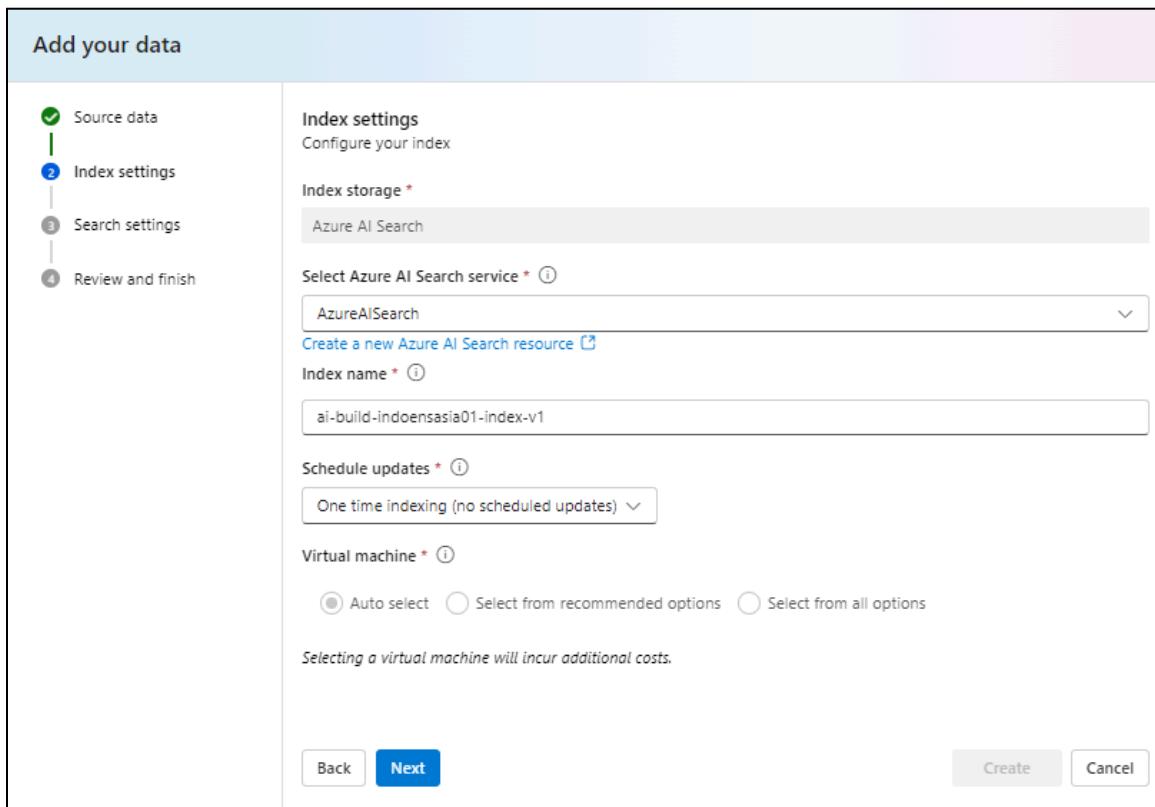
**Schedule updates \*** ⓘ  
One time indexing (no scheduled updates)

**Virtual machine \*** ⓘ

Auto select  Select from recommended options  Select from all options

Selecting a virtual machine will incur additional costs.

Back **Next** Create Cancel



- v) Select the checkboxes to **Add Vector Search** and select an embedding model from the dropdown, it should look like “**openai-tigeranalytics-<number>**”. Click **next**.

**Add your data**

Source data  
Index settings  
Search settings  
Review and finish

**Configure search settings**  
Adding vector search supports: Hybrid (vector + keyword search), Hybrid + Semantic (most accurate search results for generative AI applications), Vector, Semantic and Keyword retrieval. Hybrid will be set as default and can be changed at inference time in the playground. Not adding vector search supports: Keyword and Semantic retrieval. Keyword will be set as default and can be changed at inference time in the playground. Adding vector search requires an Azure OpenAI embedding model. [Learn more](#)

**Vector settings**  
 Add vector search to this search resource

Azure OpenAI resource \* ⓘ  
openai-tigeranalytics-101

Azure OpenAI embedding model, **text-embedding-ada-002 (Version 2)**, will be deployed if not already. Adding vector embeddings will incur usage to your account. [View pricing](#)

Back **Next** Create Cancel

vi) Review the steps, click **Create**, and wait a few minutes to finish.

**Add your data**

Source data  
Index settings  
Search settings  
Review and finish

**Review and finish**  
Review the configurations you set for your index

Index name  
ai-build-indoensasia01-index-v1

Index storage  
Azure AI Search

Azure AI Search connection  
AzureAISearch

Include vector settings  
Yes

Embedding model connection  
openai-tigeranalytics-101

Schedule  
OneTime

Compute  
Serverless compute (Auto select)

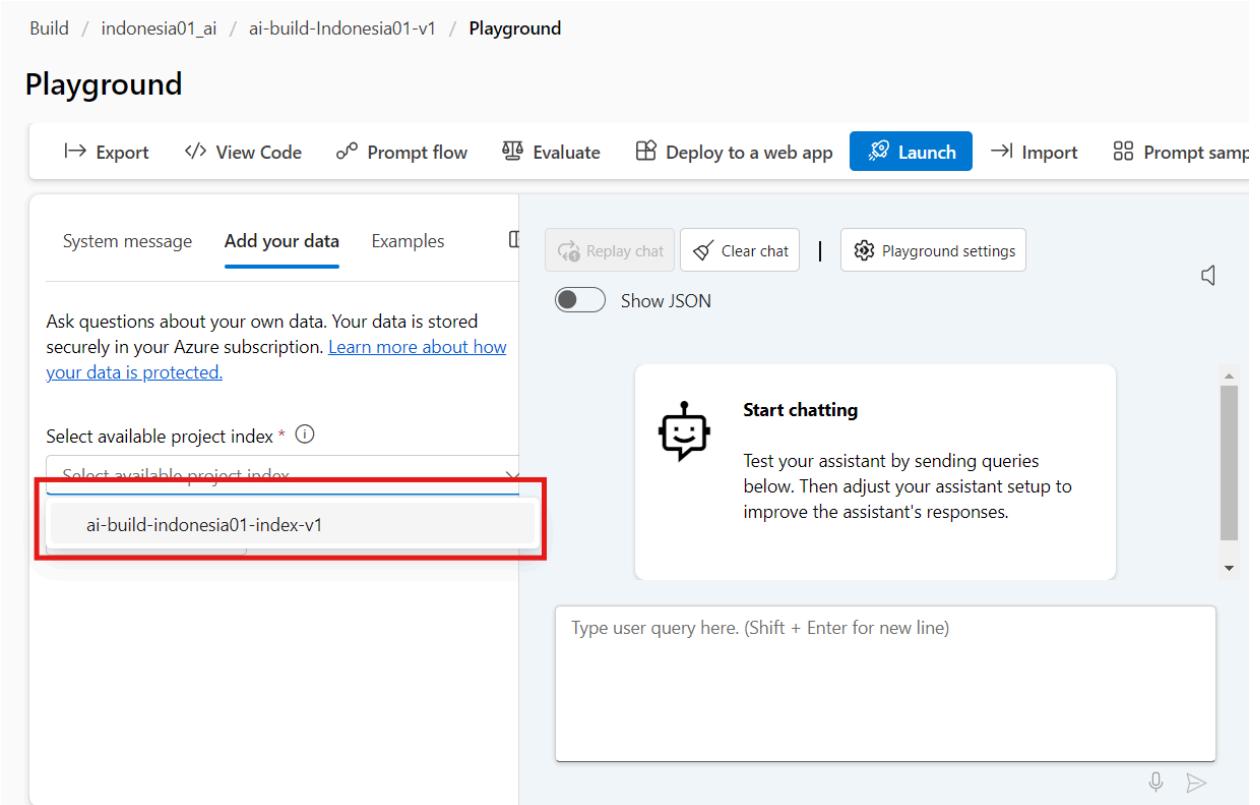
Back **Create** Cancel

**Note:** After returning to the playground, you can see the index creation progress in the “Add your data” tab. It can take **10-15 mins** to finish. You can explore other features on the playground

during that time or take a small break. If it takes too long to load the indexed data, please try the following steps:

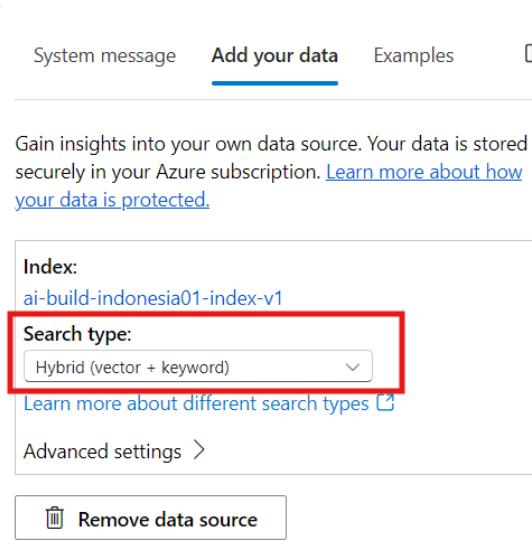
You can refresh the browser, Clear cookies, Close the browser, log in again, and navigate back to the project you created. If that does not work, contact one of our coordinators.

- vii) In the playground, select the **index** you have created, for example, **ai-hack-indonesia01-index-v1**,



The screenshot shows the Azure AI Playground interface. At the top, there are several navigation links: Build / indonesia01\_ai / ai-build-Indonesia01-v1 / Playground. Below the navigation is a toolbar with icons for Export, View Code, Prompt flow, Evaluate, Deploy to a web app, Launch (which is highlighted in blue), Import, and Prompt sample. The main area has tabs for System message, Add your data (which is selected and underlined in blue), and Examples. There are also buttons for Replay chat, Clear chat, and Playground settings. A toggle switch labeled "Show JSON" is turned off. On the right, there's a "Start chatting" section with a robot icon and a text input field for user queries. The "Add your data" section contains a dropdown menu titled "Select available project index \*". The dropdown shows "Select available project index" and "ai-build-indonesia01-index-v1", with the latter being highlighted by a red box.

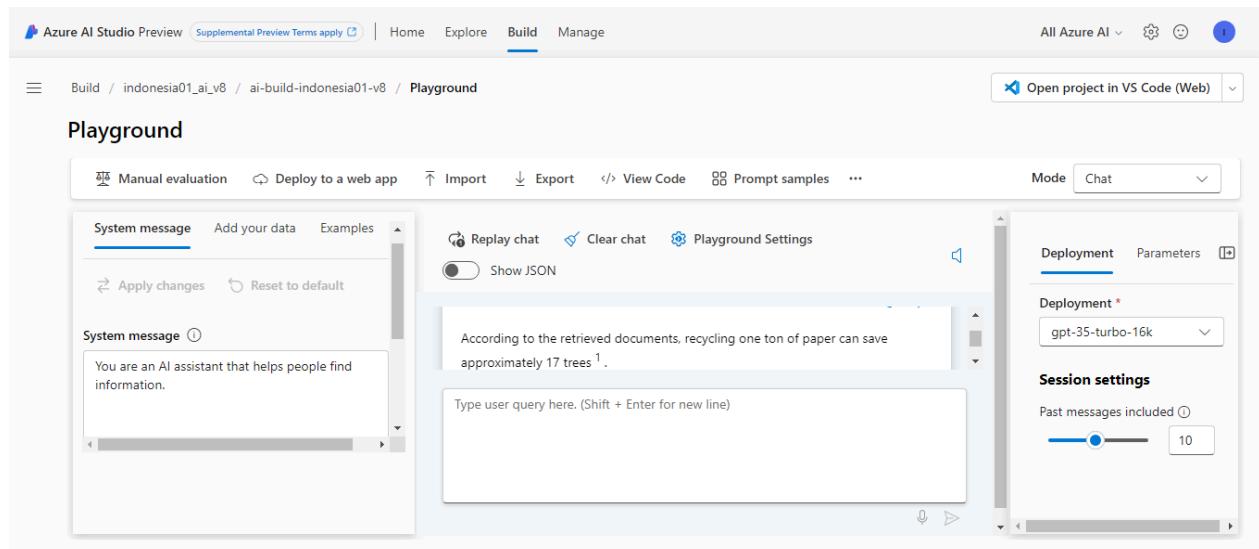
- viii) Under the **Search type** dropdown, select **Hybrid (vector + keyword)**



This screenshot shows the "Add your data" configuration page. It includes sections for "System message", "Add your data" (selected), and "Examples". A message states: "Gain insights into your own data source. Your data is stored securely in your Azure subscription. [Learn more about how your data is protected](#)". Below this, there's an "Index:" dropdown containing "ai-build-indonesia01-index-v1". A "Search type:" dropdown is highlighted with a red box and set to "Hybrid (vector + keyword)". There's also a link "Learn more about different search types" and an "Advanced settings" link. At the bottom is a "Remove data source" button.

## 4. Test the connection

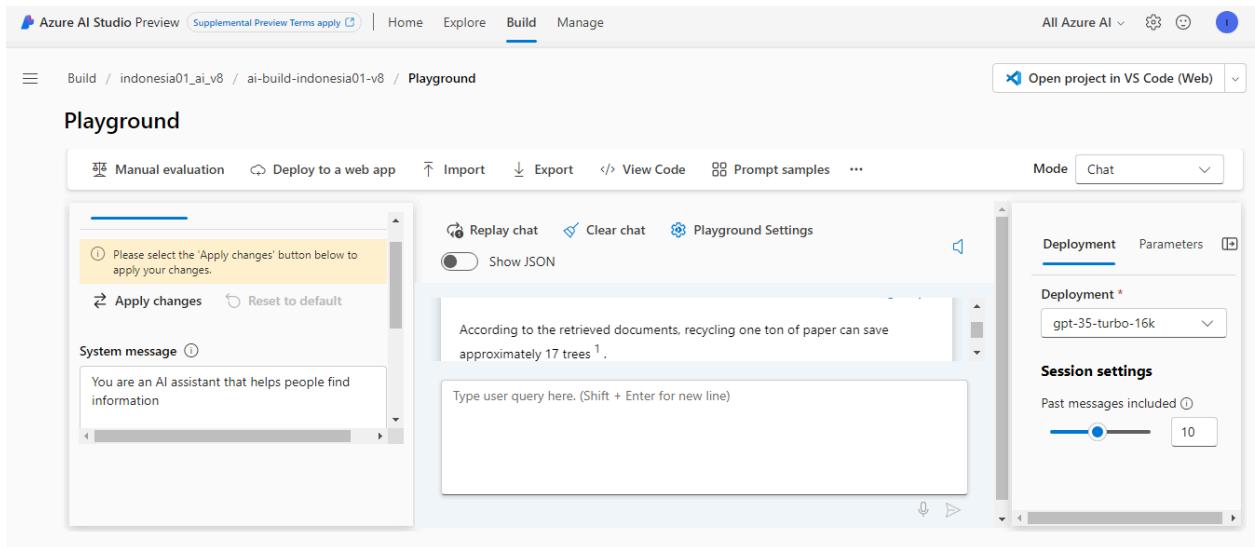
1. Ask a question in the playground chat box (Ex: "How many trees can be saved by recycling one ton of paper?")
2. If you get a system response, the connection is set, and we're good to go to the next step.



3. The system is now set up to respond to general user questions. We can add our data in the backend to unlock the power of Azure AI studio, which is the ability to generate responses from the sources provided by the user.

## 5. Prompt

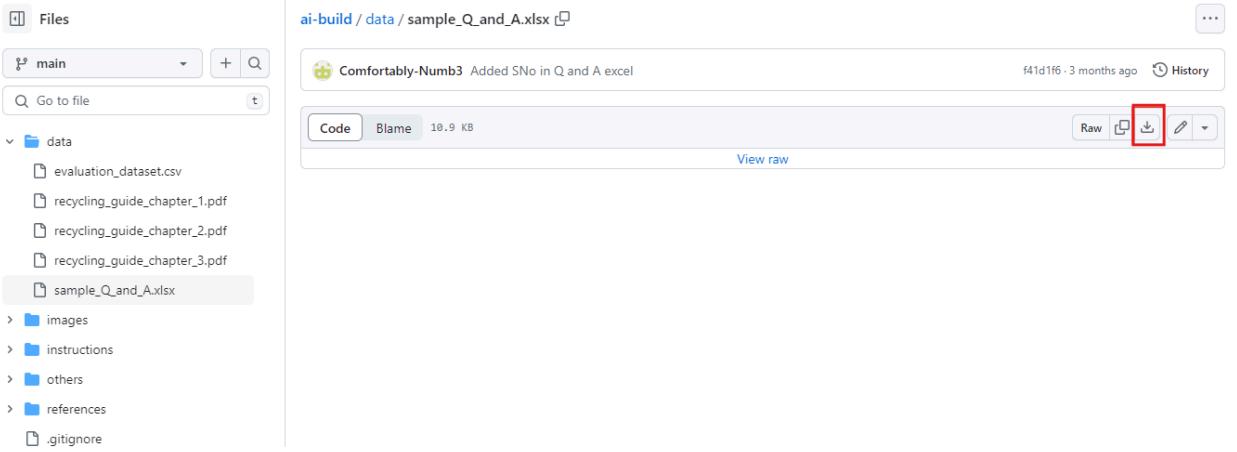
1. Define the system prompt in the prompt box and click on **Apply changes** to reflect the changes in the system responses.



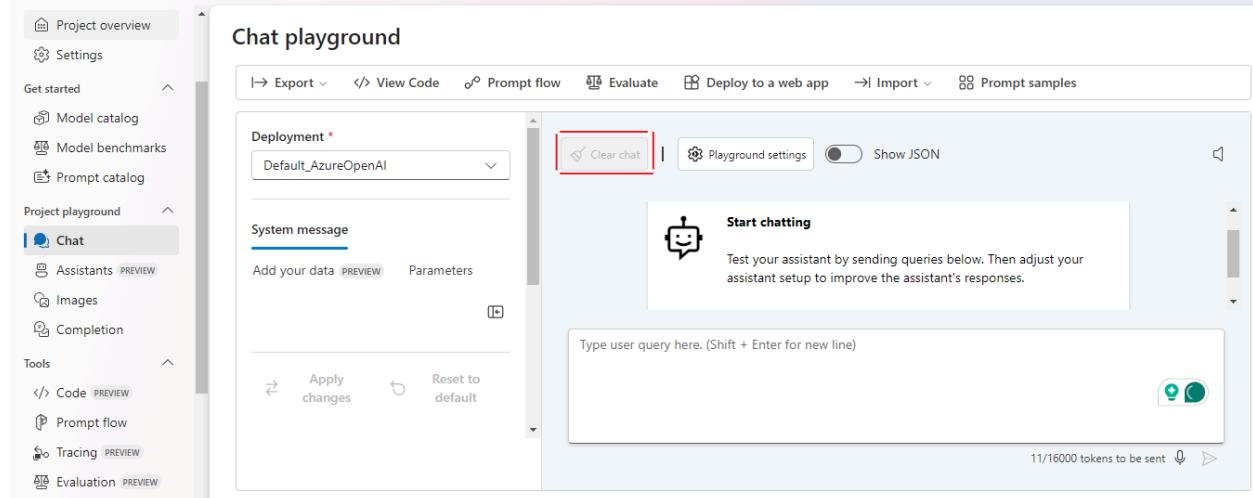
2. Navigate to the left pane, you can find the sample prompts in the **Prompt catalog** section under **Get Started** section.

## 6. Start asking questions

1. Check on the responses from the data and ask questions. Some sample questions on the data can be found [here](#) (Download this **sample\_Q\_and\_A.xlsx** in the **data** folder to view sample questions).

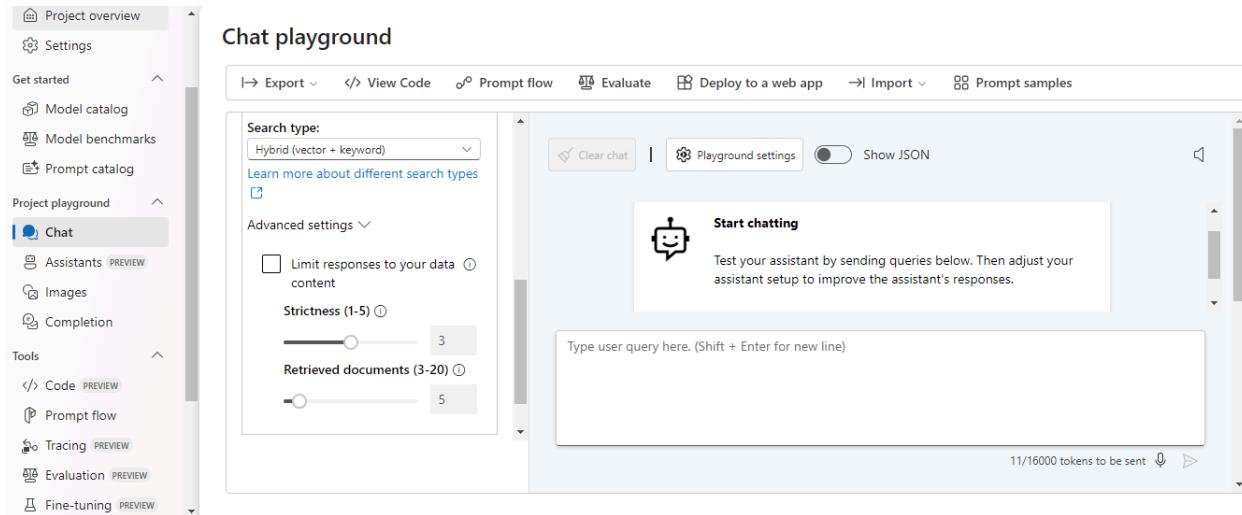


2. When you want to ask questions from the sample data, Navigate back to the left pane. Go to **Project playground -> Chat**. Type your question in the text box and wait for a response. Please click the **Clear Chat** button in the chat box before asking an unrelated follow-up question.



## 7. Tweak the parameters

1. Suppose you want the responses to come not only from the source data alone but also from the external knowledge of the Azure OpenAI models. In that case, you can click on **Advanced Settings** under **Add your data** tab and uncheck **Limit responses to your data content**.
2. **Strictness** helps you set the threshold for relevant documents, and **Retrieved documents** help you set the number of chunks/documents that will be retrieved to answer the questions.



3. Please check this guide on [how to generate text with Azure OpenAI Service - Azure OpenAI | Microsoft Learn](#)
4. Tweak the model parameters from the **Parameters** tab. You can find the definition of the parameters [here](#).
5. Make changes to the prompts to change the response to how you like (for example, we can add to give the response in bullet points or a happy tone). Remember to click on **Apply Changes** after changing the system prompt.
6. Under the **system message** tab, Define variables by selecting **Variable** in the **Add section** drop-down for easy prompt changes. Then, enter a name for the variable. Click **Apply Changes**. For example, tone/style can be set as a variable in the prompt whose value can be changed easily across different runs. This should be set in the system message as **{{format}}** in the prompt as shown below.

### Chat playground

System message ⓘ

You are an AI assistant that helps people find information.

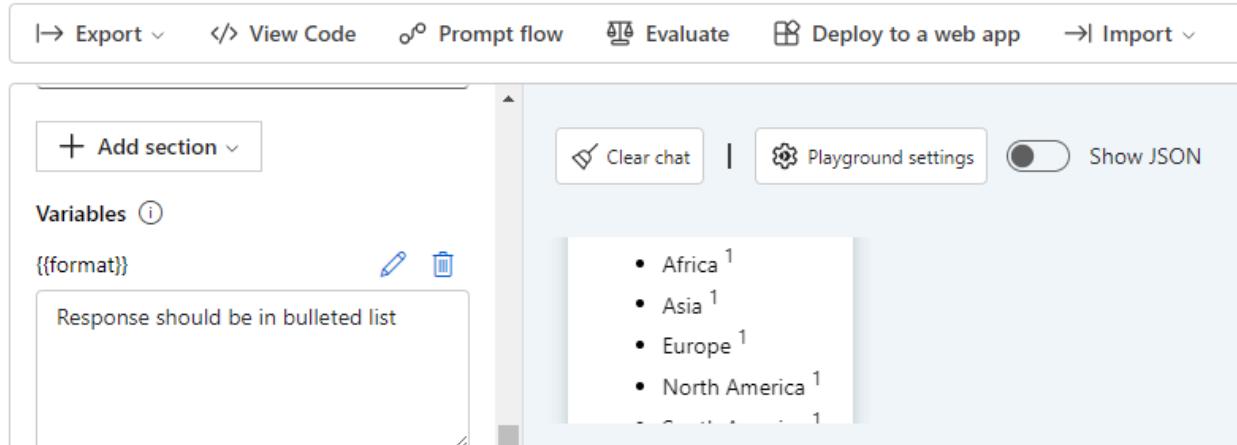
Output {{format}}

Apply changes    Reset to default

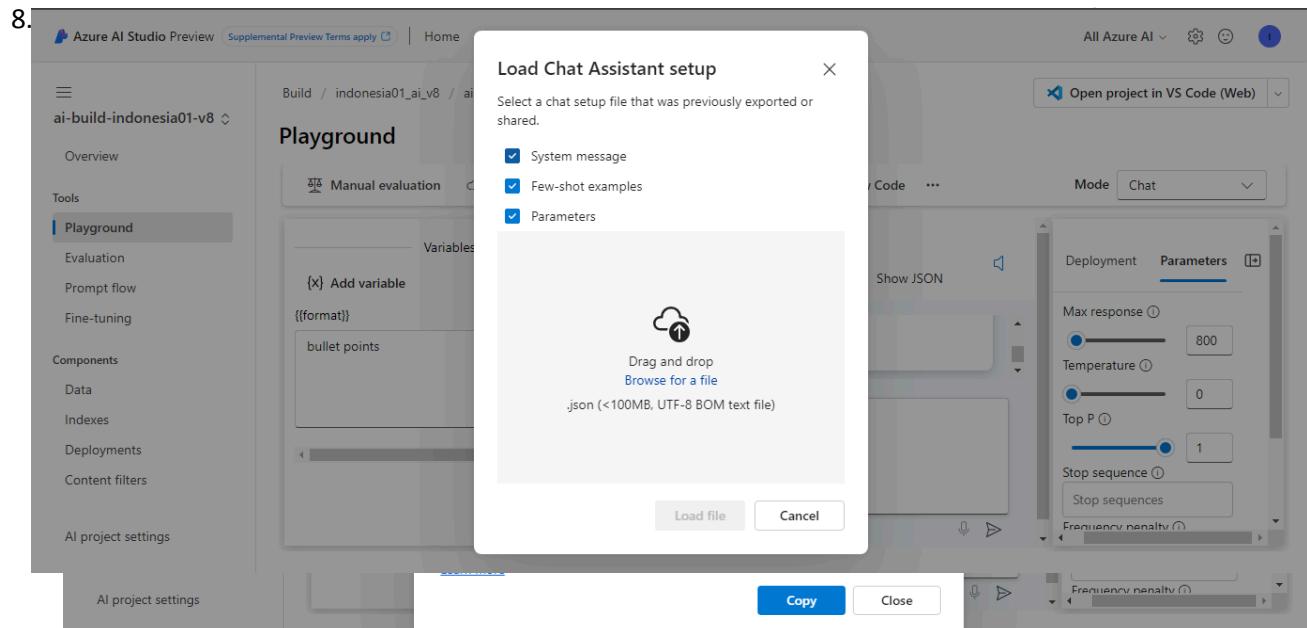
Name 5 continents in the world!

Type user query here. (Shift + Enter for new line)

## Chat playground

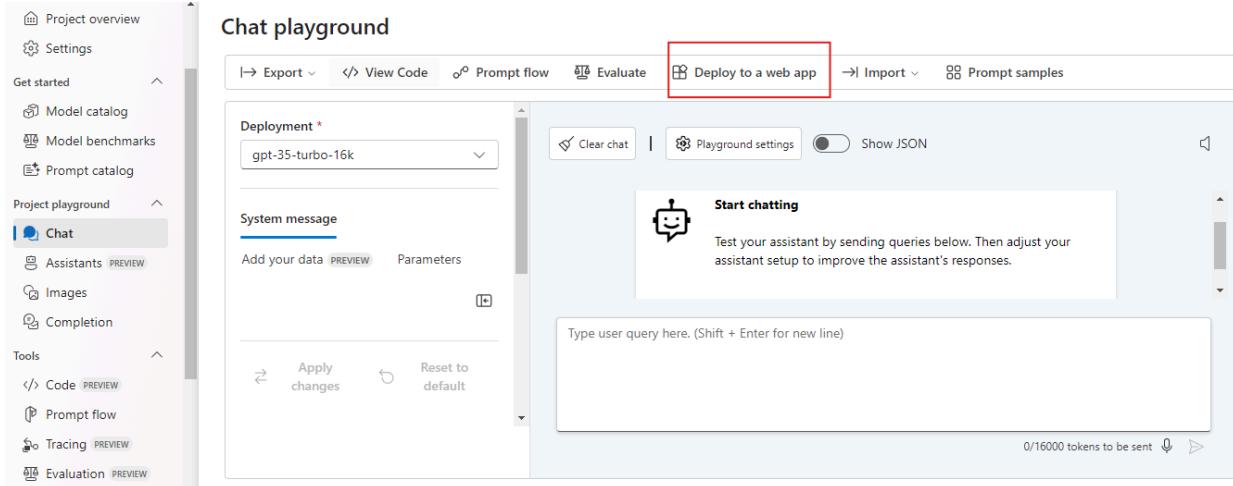


7. You can also **import** and **export** the chat settings (prompts, examples, parameters) to a JSON. You can try the same using this JSON. This feature helps share your workspace settings while working as a team.



## 8. Deploy the model as a web application

1. Click on “Deploy to a Web App”.



2. Enter a name for the web app in the format "**user-id-web-app**." Example "**Indonesia01-web-app**"
3. Select the subscription that gets populated by default, for example, "**Tiger analytics-01**", resource group as the one you have created during the creation of the AI hub. generally, it will be of the format "**rg-<userid>ai**." For example, "**rg-Indonesia01ai**" and its location are "**Australia East**."
4. Select the "**Basic B1**" pricing plan.
5. Check all "**Enable chat history in the web app**" and review the "View pricing" hyperlinks if needed.
6. Click the **deploy** and wait for a few minutes (this might take ~10 minutes) till the deployment is complete
7. If you get a message saying Deployment failed, please reload the page and try again
8. **Note: deploy the app first and then do the prompt engineering**

Pick your configurations to deploy a web app. [Learn more about web apps](#)

Create a new web app  Update an existing web app

Name ① \*

natarajansubram-web-app

Subscription ① \*

Tiger analytics-01

Resource group ① \*

rg-natarajansubramai

Location ① \*

Australia East

Pricing plan ① \*

Basic (B1)

Enable chat history in the web app

Enabling chat history will incur CosmosDB usage to your account [View Pricing](#)

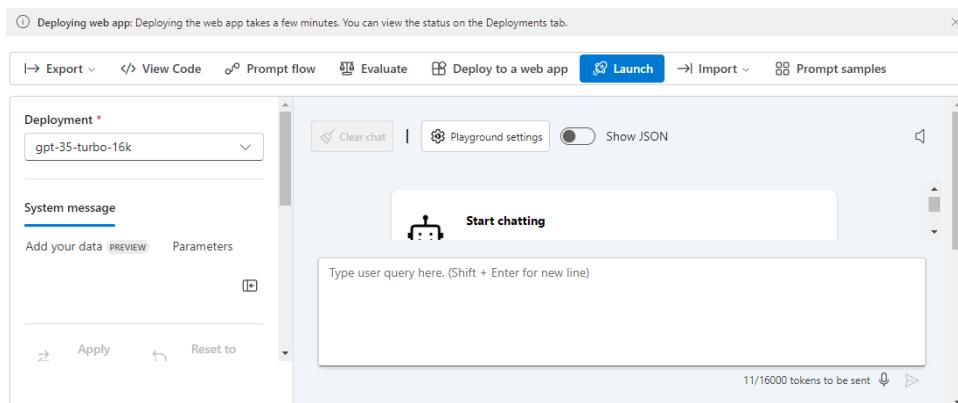
Web apps will incur usage to your account [View Pricing](#)

Deploy

Cancel

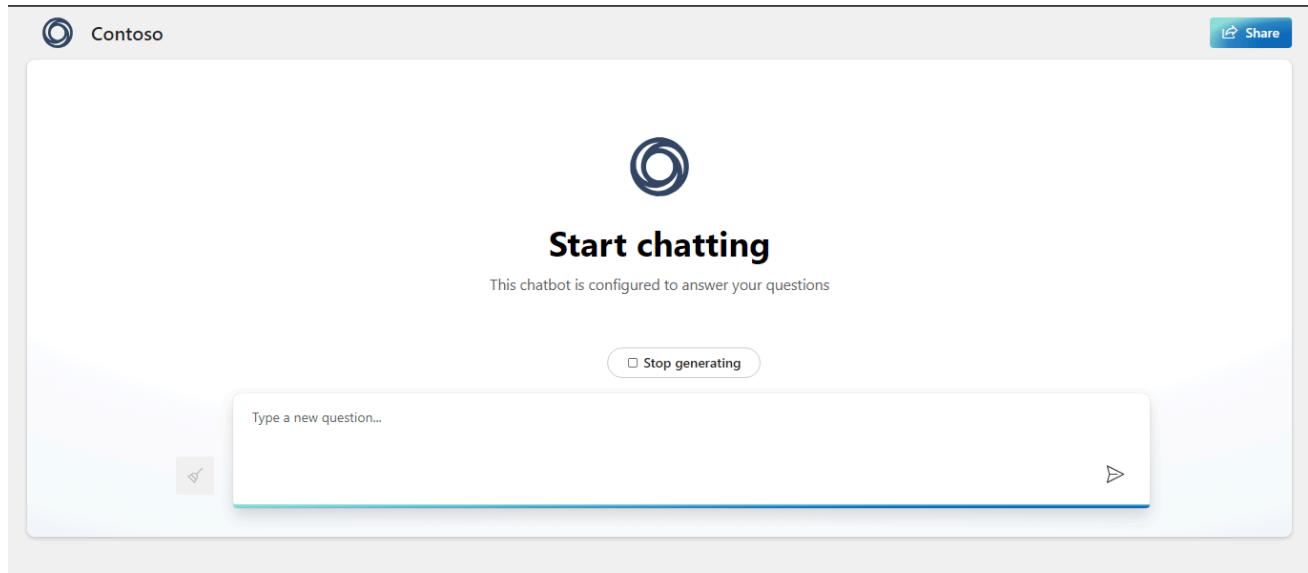
- Deployment can take up to 10 minutes. When it's done, the launch button will become enabled. click on the “**Launch**” button

Chat playground



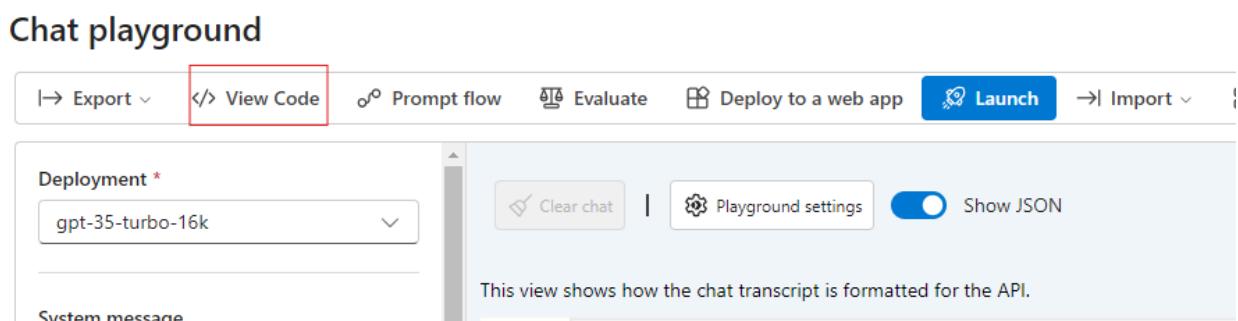
- Once you click on the **Launch** button to open the URL, you will be asked to “Accept” the invitation. Please do and proceed forward. If you see an error about the “Authentication,” please wait for a few minutes, and you will be able to see a sample web application, as shown in the screenshot below.

You can start asking questions using the web application. People accessing your AI Studio project can also access the application via URL and ask questions. All the logs, user questions, and system responses are stored in a database.



## 9. Additional Playground Features

1. You can also view the entire project as code by clicking the “**View Code**” tab in the playground as shown below. It will then open a new dialog box that shows the python code implementation.



## Sample Code

You can use the following code to start integrating your current prompt and settings into your application.

Endpoint ⓘ

https://openai-tigeranalytics-101.openai.azure.com/



API key ⓘ

.....



You should use environment variables or a secret management tool like Azure Key Vault to prevent accidental exposure of your key in applications.

[Learn more about setting up an environment](#)

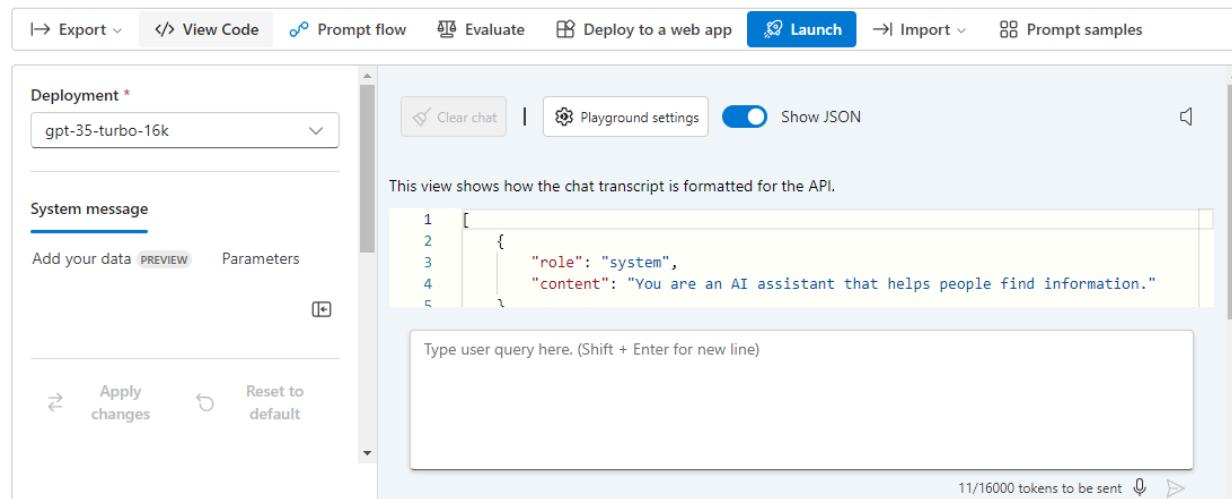
python ▾

```
1 import openai, os, requests
2
3 openai.api_type = "azure"
4 openai.api_version = "2024-02-15-preview"
5
6 # Azure OpenAI setup
7 openai.api_base = "https://openai-tigeranalytics-101.openai.azure" ▾
```

Close

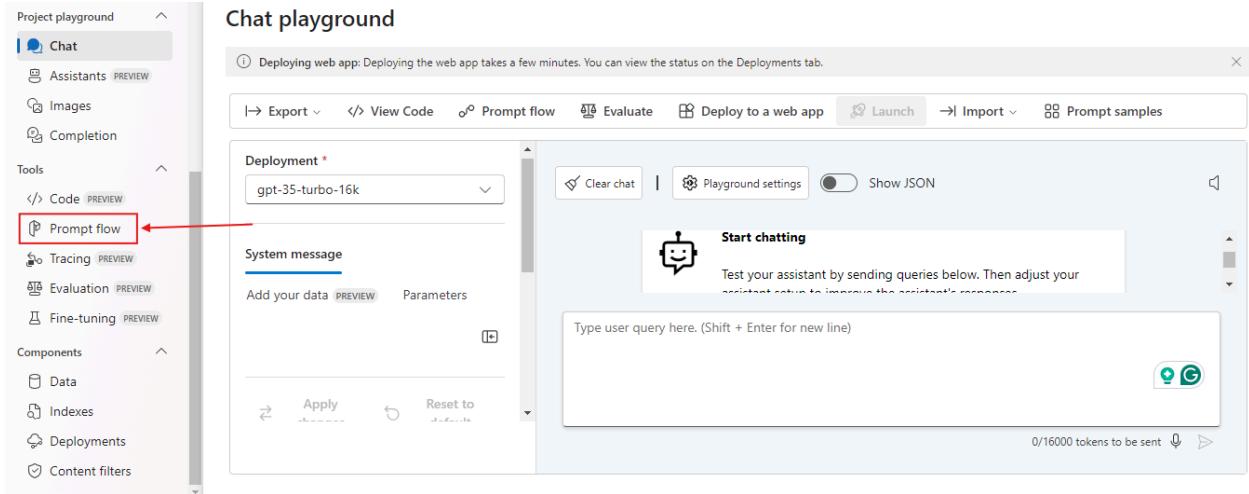
2. You can look at the JSON of the chat transcript. This is formatted for the API calls.

## Chat playground

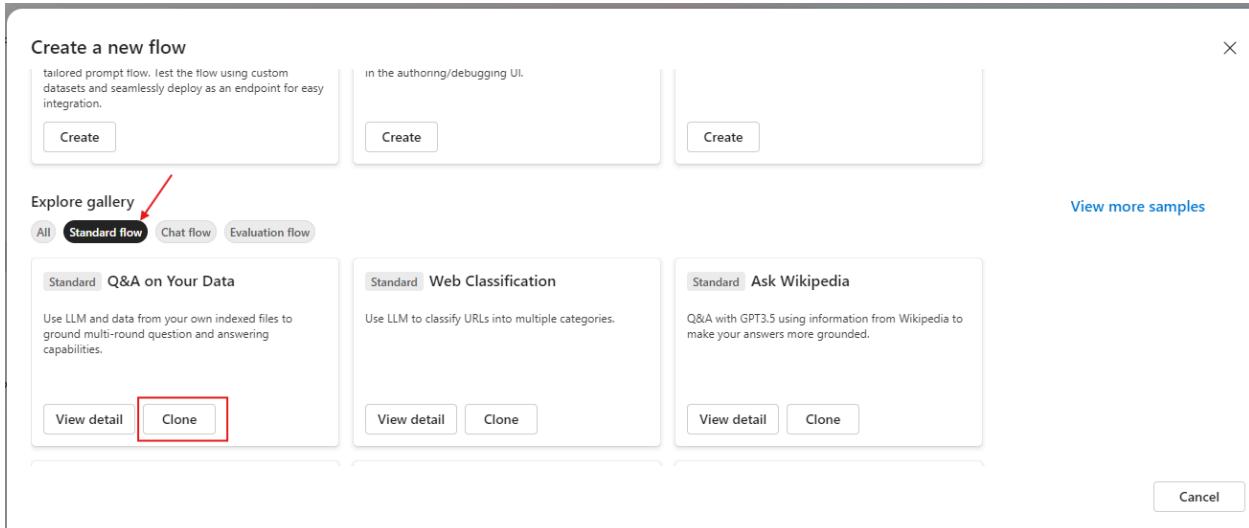


## 10. Prompt flow

1. Navigate to the left-side pane and click on **Prompt flow** under **Tools**. Then click on the “Create” button to create a new prompt flow.



- Now you will find specific types of flows to be made which you can clone and explore for further understanding. For our example, let's select "**Standard flow**" under **Explore gallery**, now in the "**Q&A on Your Data**" section click on "**Clone**".



- Then a new popup window will open, then provide a folder name like this ai-hack-<userid>-flow-<version> (eg. ai-hack-yourname-flow-v1) and click on **Create**. This will take ~ 1 minute to load.

## Create a new flow

X

The flow code files are stored in a specific folder within your workspace file share storage. This folder name can be customized according to your preferences.

Location to store flow \* ⓘ

Users/bhavesh.pancholi/promptflow

Folder name \* ⓘ

Flow-created-on-06-13-2024-18-55-03

Create

Cancel

4. In the prompt flow-right pane, you can see the “**Graph**,” which gives you a bird's-eye view of the whole flow, and the “**Files**,” which lets you download the prompt flow files to share your work. Click on the dropdown option next to **Run** & select **Start compute session**.

The screenshot shows the Azure AI Studio interface with the 'Flow' tab selected. On the left, there's a sidebar with sections like 'Project playground', 'Tools' (including 'Prompt flow'), and 'Components'. The main area displays a flow configuration with an 'Inputs' section containing 'chat\_hist' (list type) and 'question' (string type) fields, and an 'Outputs' section with an 'answer' field set to `\${chat.output}`. In the top right, there's a 'Graph' view showing a flow from 'inputs' to a 'chat' component, which then leads to 'outputs'. A prominent red box highlights the 'Start compute session' button in the top right corner of the main panel.

5. In the **Flow** tab, you can see all the components that are a part of this flow. Collapse them to get an overview of the whole structure, as shown below.

This screenshot shows the same Azure AI Studio interface as above, but with the components expanded. The 'Inputs' and 'Outputs' sections are now collapsed, revealing the internal components: 'echo' and 'joke'. The 'joke' component is highlighted with a red box. The 'Graph' view on the right shows the expanded structure: 'inputs' connects to 'echo', which then connects to 'joke', which finally connects to 'outputs'.

6. Setup the OpenAI connection in the “`answer_the_question_with_context`” component in the **Flow** tab. Click on the drop down next to **Connection** and select `openai-tigeranalytics-<number>`. In

**deployment\_name** select **gpt-35-turbo-16k**. In **response\_format** select **{"type": "text"}**.

The screenshot shows the 'Flow' interface with the 'answer\_the\_question\_with\_context' component selected. The 'deployment\_name' dropdown is set to 'gpt-35-turbo-16k'. The 'response\_format' dropdown is set to '["type": "text"]'. The Graph view on the right shows the component connected to a 'Prompt\_variants' node and an 'outputs' node.

7. Now scroll above to the **lookup** section, under **Inputs** click on **Value** in **mlindex\_content**.

The screenshot shows the 'Flow' interface with the 'lookup' component selected. The 'Value' field for 'mlindex\_content' is highlighted with a red box and an arrow pointing to it. The Graph view on the right shows the 'inputs' node connected to the 'lookup' component, which then connects to the 'generate\_prompt\_context' component.

8. Now a popup window will open, then fill the following details and click on "Save".

For **index\_type** in **Value** dropdown select **Azure AI Search**.

For **acs\_index\_connection** select **AzureAISearch**

For **acs\_index\_name** select **ai-hack-<userid>-index-<version>**

For **acs\_content\_field** select **content**

For **acs\_embedding\_field** select **contentVector**

For **acs\_metadata\_field** select **meta\_json\_string**

For **semantic\_configuration** select **azureml-default**

For **embedding\_type** select **Azure OpenAI**

For **aoai\_embedding\_connection** select **openai-tigeranalytics-101**

For **embedding\_deployment** select **text-embedding-ada-002**

Generate

| Name                   | Type            | Value                  |
|------------------------|-----------------|------------------------|
| index_type             | string          | Azure AI Search        |
| acs_index_connection   | Azure AI Search | AzureAISeach           |
| acs_index_name         | string          | ai-hack-rajeshdash-v01 |
| acs_content_field      | string          | content                |
| acs_embedding_field    | string          | contentVector          |
| acs_metadata_field     | string          | meta_json_string       |
| semantic_configuration | string          | azureml-default        |

**Save** **Cancel**

9. Then in the **lookup** section, under **Inputs for query\_type** select **Hybrid (vector + keyword)**.

The screenshot shows the Flow interface with the 'lookup' component selected. On the left, the 'Inputs' section is expanded, showing the following configuration:

| Name            | Type   | Value                                    |
|-----------------|--------|--|
| mlindex_content | string | embeddings: api_base: https://openai.com |
| queries         | object | \$(inputs.question)                      |
| query_type      | string | Hybrid (vector + keyword)                |
| top_k           | int    | 2  |

On the right, the 'Graph' pane displays the flow structure:

```

graph TD
    lookup[lookup] --> generate[generate_prompt_context]
    generate --> variants[Prompt_variants]
    
```

10. Then in the **Inputs** section, in **Value**, enter some text. For example: **How many trees can be saved by recycling one ton of paper?** Then on the top right, click **Run**.

The screenshot shows the AI Platform interface with a flow named "ai-hack-bhavesh-flow-v1". The top bar includes options like Clone, Save, Deploy, Evaluate, and a Compute session status. The main area has tabs for Flow, Files, and Graph. The Flow tab shows an "Inputs" section with a table:

| Name     | Type   | Value  | Action  |
|----------|--------|--|---------|
| question | string | How many trees can be saved by recycling one t | [trash] |

Below this is an "Outputs" section with a "lookup" node selected. The Graph tab shows a diagram with a blue box labeled "inputs" connected by a blue arrow to a "lookup" node.

11. Now a popup window will open and select “**Use default variant for all nodes**” and click on **Submit**.

## Submit flow run

X

You have set variants for following nodes. The flow will generate a separate run for each variant. You can select a node with variants to run. Or select "Use default variant for all nodes" to avoid generating separate runs, so the default variant of each node is used.

Select the LLM node with variants that you want to run.

Select a node to run variants    Use default variant for all nodes

| Name            | Default variant |
|-----------------|-----------------|
| Prompt_variants | variant_0       |

**Submit**

**Cancel**

12. The run will be complete & you will see the results. To see the output, click on **View trace** in the green notification bar.

variant\_0 Run completed:

**View trace**

ai-hack-bhavesh-flow-v1

Flow

Outputs

Input Output Trace Logs

Duration 2.22s Completed View full output

```

{
  "output": [
    {
      "text": "Title: Recycling Guide Chapter 2.pdfChapter 2: Benefits of Recycling\nRecycling offers a wide range of benefits to the environment, economy, and society, supported by factual numbers and data.\nEnvironmental Benefits: From an environmental perspective, recycling significantly reduces the environmental footprint associated with resource e"
    }
  ]
}
  
```

generate\_prompt\_context

Files

Graph

outputs

Input & Output Raw JSON Exception 0 Logs

### Input

```
{
  "question": "How many trees can be saved by recycling one ton of paper?"
}
```

### Output

```
{
  "output": "According to the Environmental Protection Agency (EPA), recycling one ton of paper saves approximately 17 trees. (Source: Recycling Guide Chapter 2.pdf)"
}
```

## 11. Evaluation

- To create a new evaluation flow, navigate to the playground's left pane. Under the **Tools** section, click **Evaluation** (page appears as shown) and click the **new evaluation** button.

The screenshot shows the Azure AI Studio interface with the path: Azure AI Studio / All hubs / natarajansubramai / All projects / ai-hack-natarajansubram01-v1 / Evaluation. The main title is "Assess and compare AI application performance". Under "Metric evaluations PREVIEW", there is a section titled "Risk and safety check" with four bar charts. The charts show defect rates for different content types across severity scales.

| Metric                     | Defect rate (%) | Severity Scale              |
|----------------------------|-----------------|-----------------------------|
| Violent content            | 6.00%           | Very low, Low, Medium, High |
| Self-harm-related content  | 2.00%           | Very low, Low, Medium, High |
| Sexual content             | 6.00%           | Very low, Low, Medium, High |
| Hateful and unfair content | 12.00%          | Very low, Low, Medium, High |

2. Select the appropriate scenario to evaluate. Here, we will determine the first scenario: **Question and answers with context**. To see a sample dataset that can be used in the evaluation, click the **Download data template for selected scenario**. We already have the data prepared. Change the Evaluation name to "**ai-hack-<userid>-evaluation-<version>**". Example **ai-hack-indonesia01-evaluation-v8**" Then click **Next**.

The screenshot shows the "Add basic information" step of the evaluation wizard. The left sidebar lists steps: 1 Basic information (selected), 2 Configure test data, 3 Select metrics, 4 Review and finish. The right panel shows the "Evaluation name" field with "ai-build-natarajansubram-evaluation-v1" and a "Download data template for selected scenario" button. Below are two options: "Question and answer with context" (selected) and "Question and answer without context". A "Select a flow to evaluate (optional)" dropdown is shown at the bottom.

3. Select **Add your dataset**. Then click on **Upload File**. Select **EvaluationDataset.csv** from the **data** folder [here](#). The preview of the top 3 rows will be shown after successful upload. Then click **Next**.

4. Select the metrics you want to use for the evaluation based on the definition.

| Performance and quality metrics curated by Microsoft   |   |  |
|--|---|--|
| <b>Groundedness</b><br>Measures how well the model's generated answers align with information from the input source. | <b>Relevance</b><br>Measures the extent to which the model's generated responses are pertinent and directly related to the given questions. | <b>Coherence</b><br>Measures how well the language model can produce output that flows smoothly, reads naturally, and resembles human-like language. |
| <b>Fluency</b><br>Measures the language proficiency of a generative AI's predicted answer.                           | <b>GPT similarity</b><br>Measures the similarity between a source data (ground truth) sentence and the generated                            | <b>F1 score</b><br>Measures the ratio of the number of shared words between the model prediction and   |

5. Select the OpenAI connection **openai-tigeranalytics-<number>**. The Deployment model name will be automatically populated as **gpt-35-turbo-16k**.

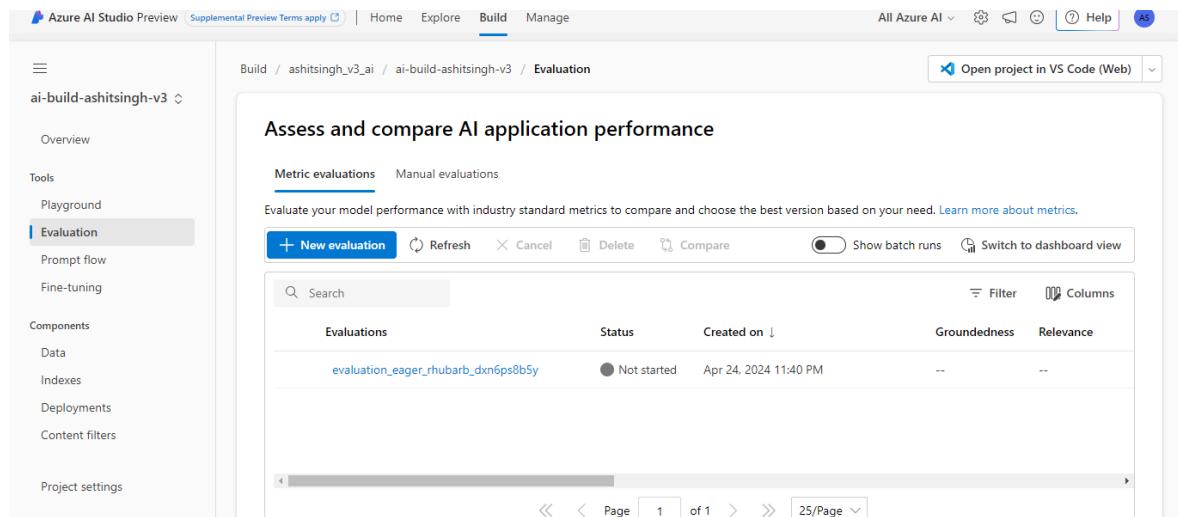
6. Map the columns from the uploaded file to the respective fields. Here it should be automatically populated. But you can change as per your choice. Click **Next**. Review the submitted information and then submit.

How does your dataset map to your evaluation input? \*

| Name         | Description   | Type   | Data source                          |
|--------------|---|--------|--------------------------------------|
| context      | The source that response is generated with respect to               | string | <code> \${data.context} </code>      |
| answer       | The response to question generated by the model as answer           | string | <code> \${data.answer} </code>       |
| question     | A query seeking specific information                                | string | <code> \${data.question} </code>     |
| ground_truth | The response to question generated by user/human as the true answer | string | <code> \${data.ground_truth} </code> |

**Back** **Next** **Cancel**

7. The mentioned evaluation name should show up in the list. It will start and be completed in some time.

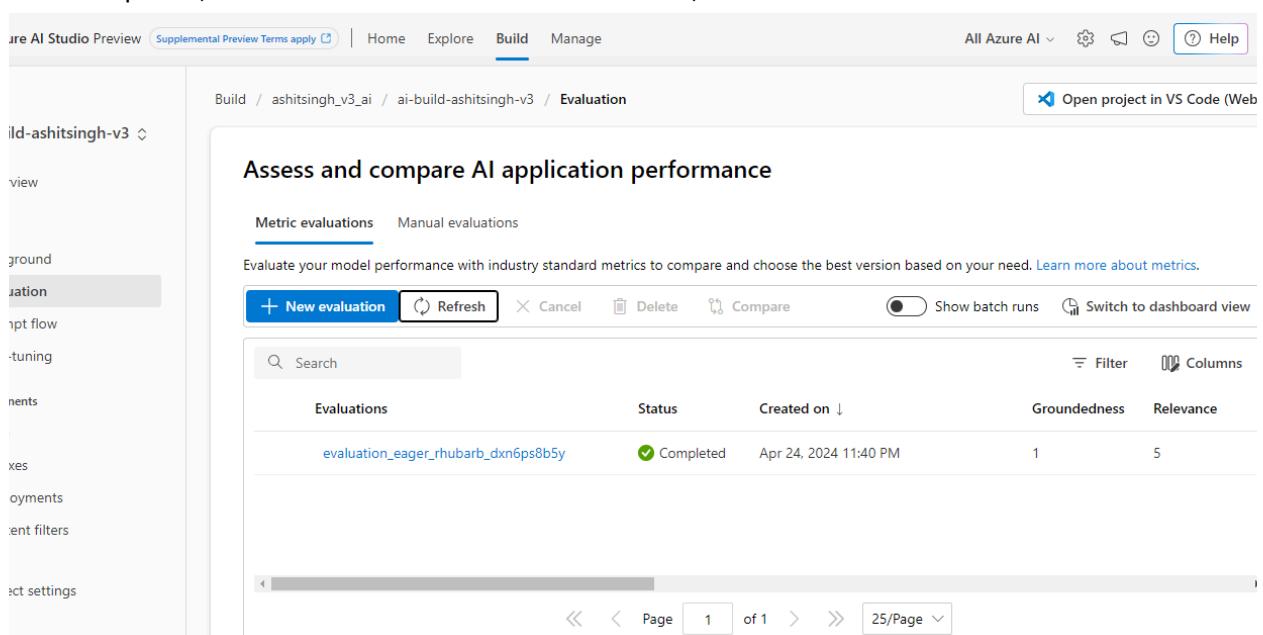


The screenshot shows the Azure AI Studio Preview interface. The left sidebar has a tree view with 'ai-build-ashitsingh-v3' expanded, showing 'Overview', 'Tools', 'Playground', 'Evaluation' (which is selected and highlighted in blue), 'Prompt flow', 'Fine-tuning', 'Components', 'Data', 'Indexes', 'Deployments', 'Content filters', and 'Project settings'. The main content area is titled 'Assess and compare AI application performance' under 'Metric evaluations'. It displays a table of evaluations:

| Evaluations                         | Status      | Created on            | Groundedness | Relevance |
|-------------------------------------|-------------|-----------------------|--------------|-----------|
| evaluation_eager_rhubarb_dxn6ps8b5y | Not started | Apr 24, 2024 11:40 PM | --           | --        |

At the bottom, there is a pagination control showing 'Page 1 of 1' and a '25/Page' dropdown.

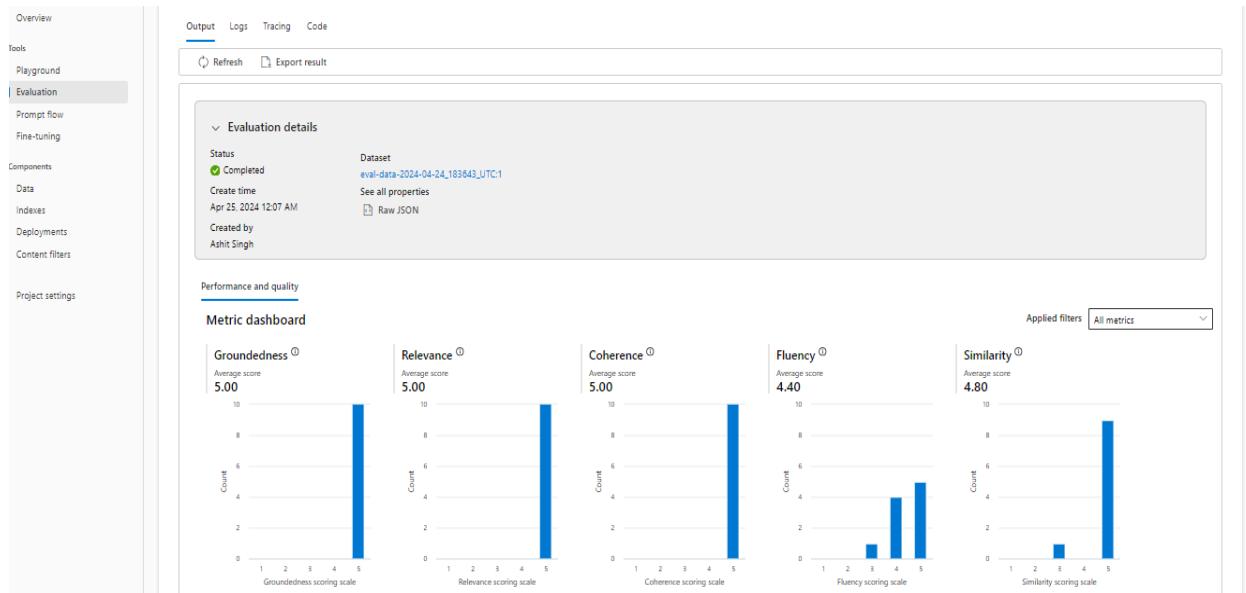
8. Once completed, click on the evaluation to see the results, as shown below.



The screenshot shows the same Azure AI Studio Preview interface as the previous one, but the evaluation status has changed. The table now shows:

| Evaluations                         | Status    | Created on            | Groundedness | Relevance |
|-------------------------------------|-----------|-----------------------|--------------|-----------|
| evaluation_eager_rhubarb_dxn6ps8b5y | Completed | Apr 24, 2024 11:40 PM | 1            | 5         |

The rest of the interface remains the same, including the navigation bar and sidebar.



You can also run **Manual Evaluations** similarly.

**Note:** The Azure service accesses used in this event are only valid during the event. Please contact your respective Microsoft account teams to follow up on continued access or if you have further questions.