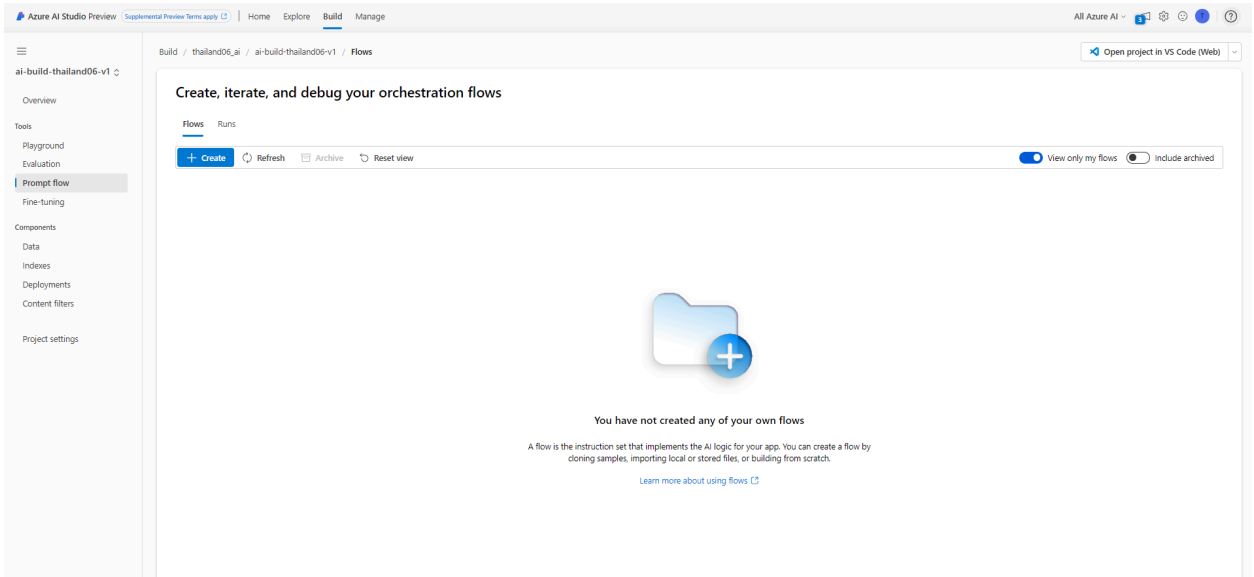
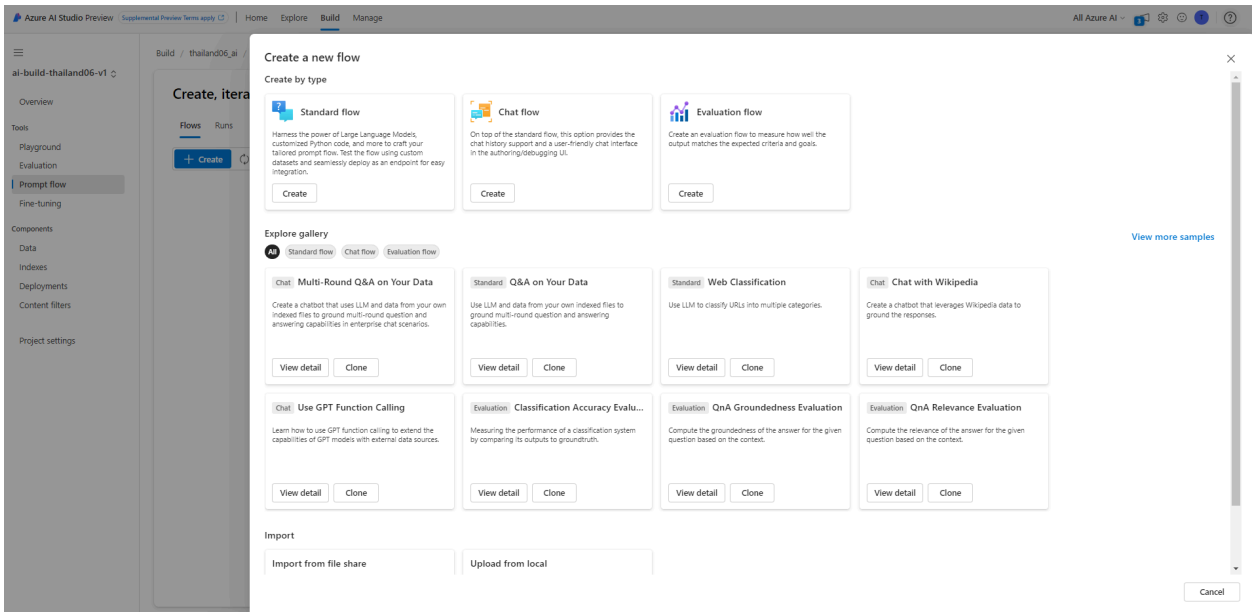
Prompt flow (Multi-Turn RAG)

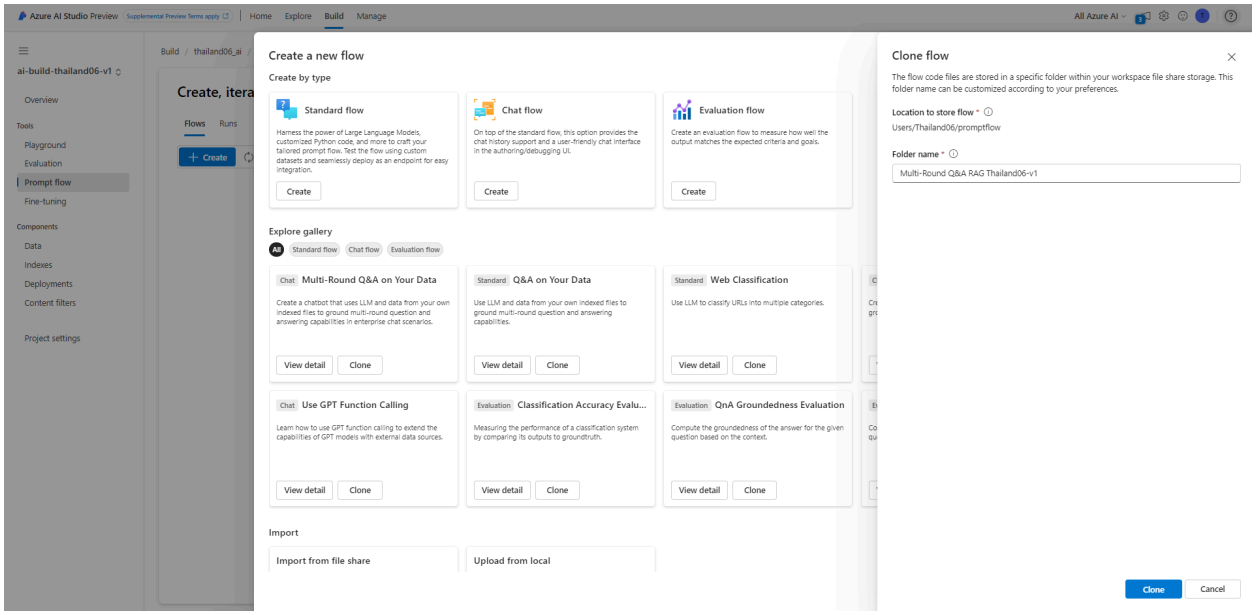
1. Go to prompt flow and click **“create”** to create a new prompt flow. You will find specific types of flows to be made and some of the “**Example Flows,”** which you can clone and explore for further understanding.



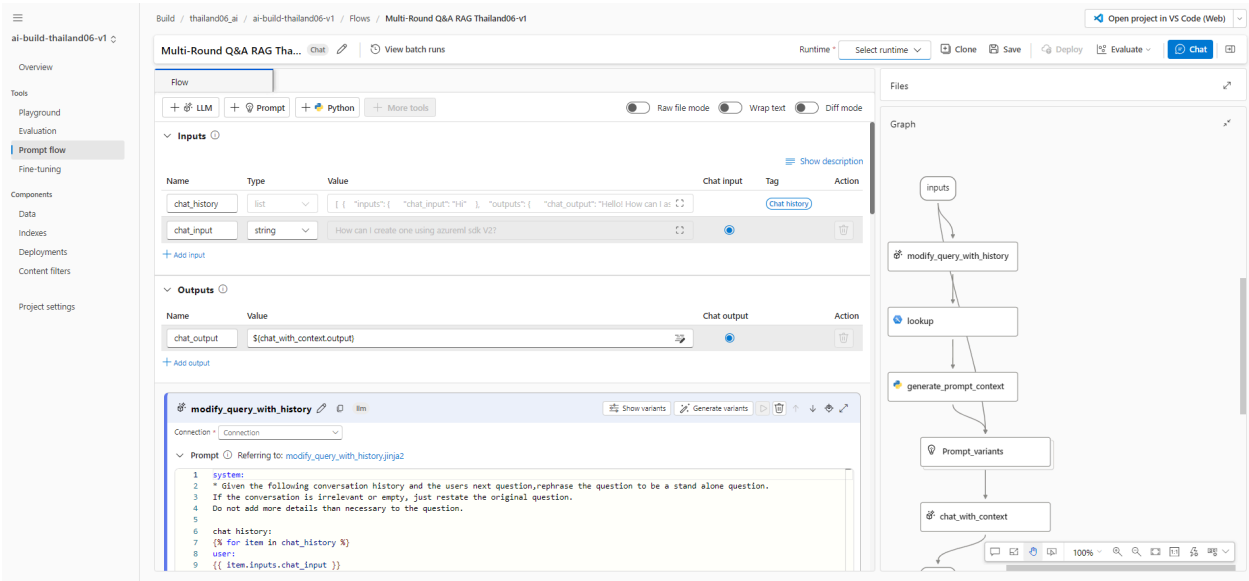
2. We will select “**Multi-Round Q&A on Your Data**” for this example. Click on “**Clone**”



3. Specify a folder name for your project. You can use the following naming convention **“Multi Round Q&A RAG” + login ID + version number.** E.g. “Multi Round Q&A RAG Thailand06-v1”

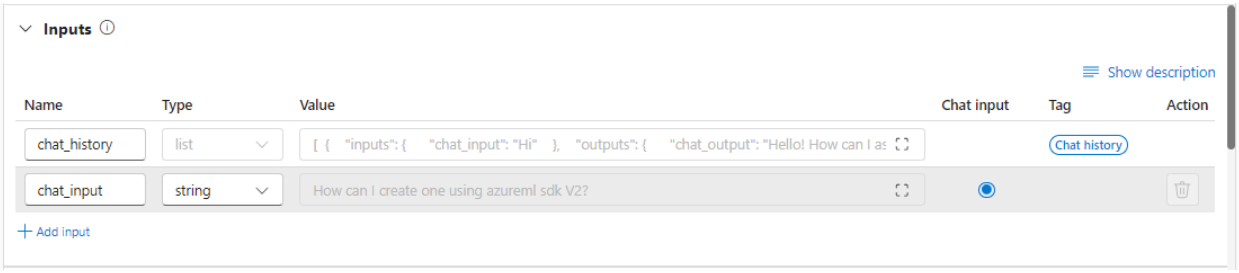


4. In the prompt flow, you can see the “**graph,”** which gives you a bird' s-eye view of the whole flow, and the “**Files,”** which lets you download the prompt flow files to share your work. Click on the dropdown option next to **Runtime** & select **Start**.



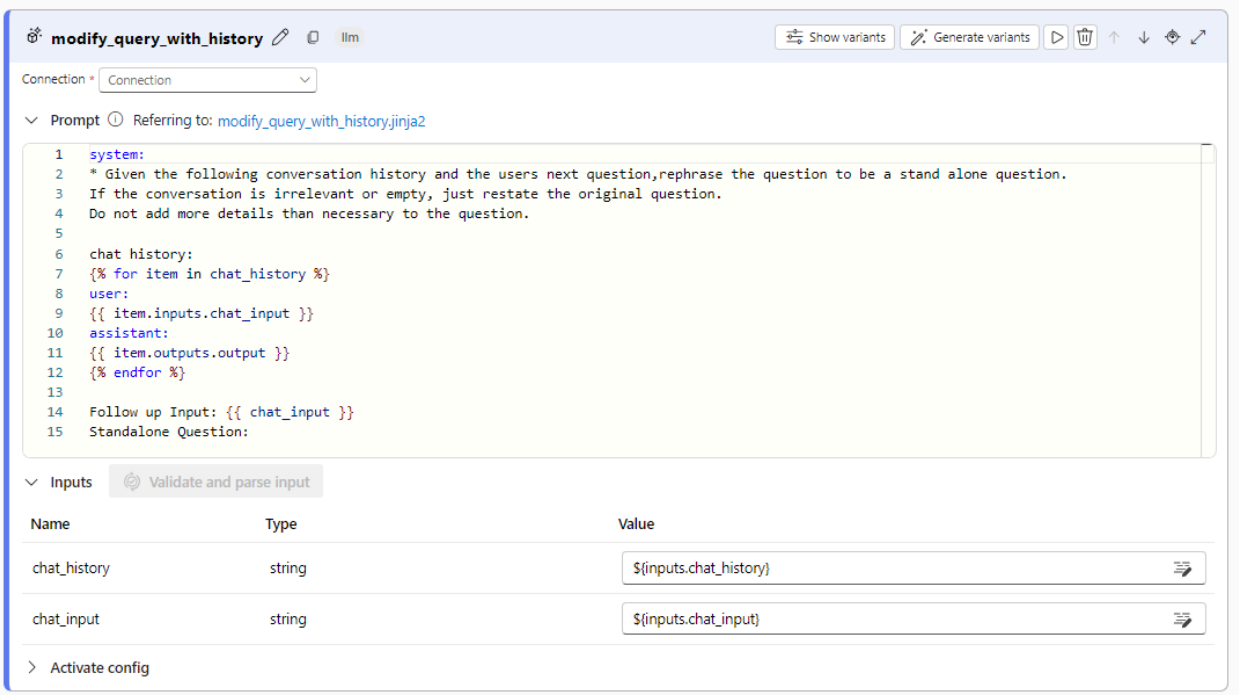
We can now start setting up the components under the **Flow** tab. Note: You can add/remove components from the flow as per the requirements.

5. Under the **Inputs** section, you can add more than one input and specify different data types for each type. In this example, **chat_input** is an input name with **string** as the data type.

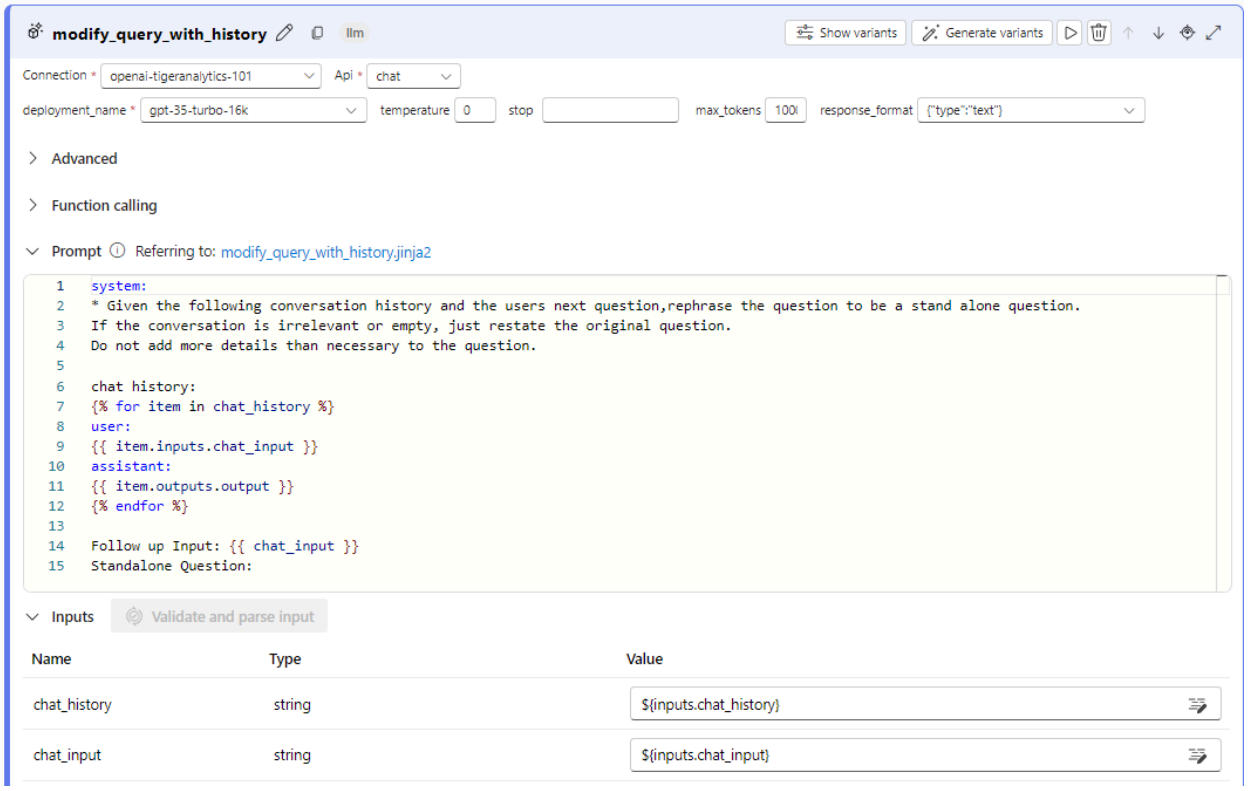


6. Under the **Outputs** section, you can specify the number of outputs and even assign values from other components of the flow. In this example, **chat_output** is the output name and **${chat_with_context.output}** is the value. Here **chat_with_context** is another component of the flow and **output** is a variable of that component.

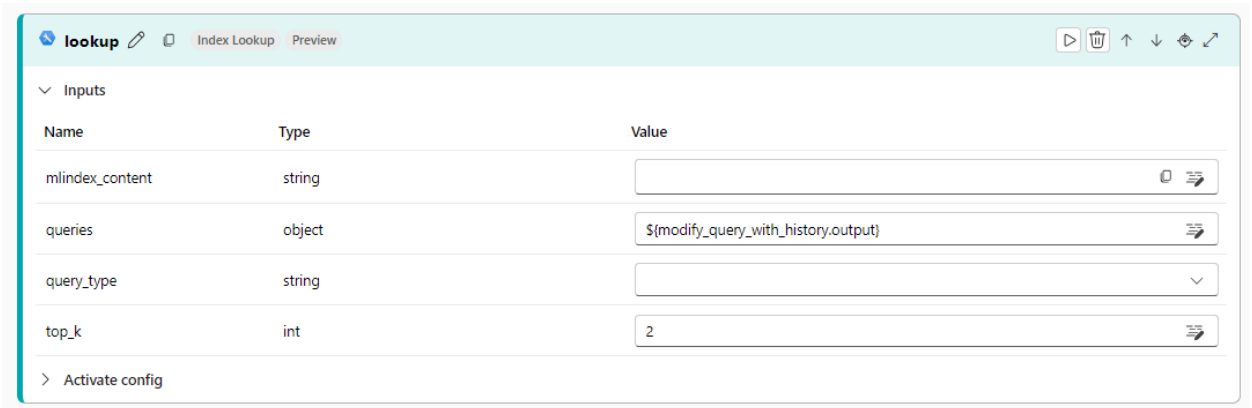
7. Under **modify_query_with_history** we would already have the prompts to create standalone questions from conversation history. We can tweak the prompts as needed.



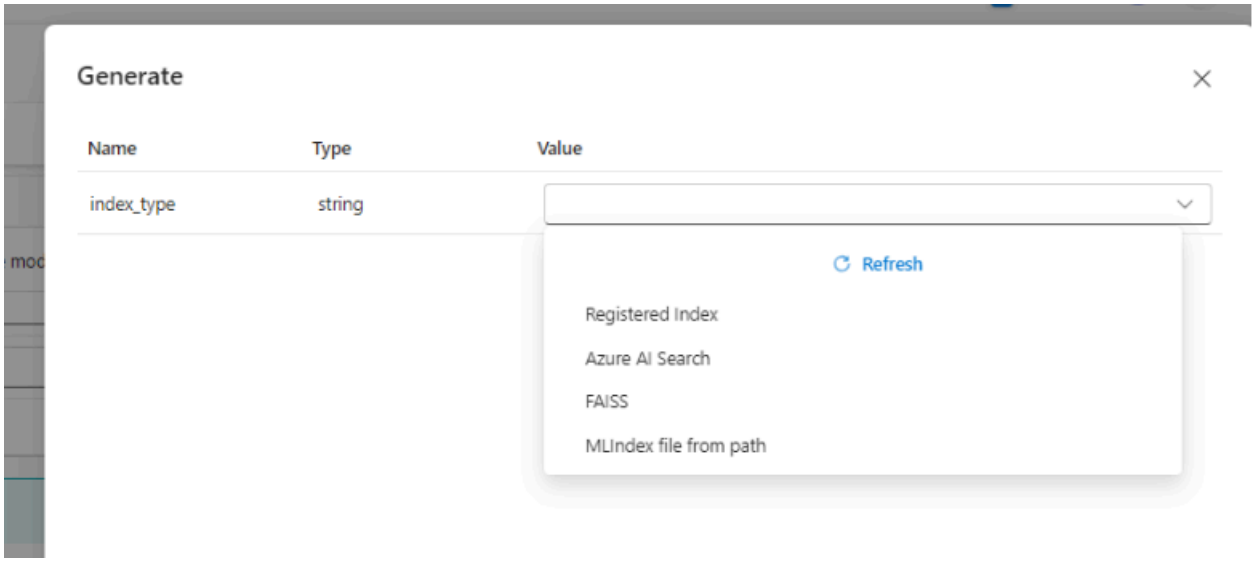
Set up the OpenAI connection in **modify_query_with_history**. Click on the drop-down next to **Connection** and select **openai-tigeranalytics-<number>**. In **deployment_name** select **gpt-35-turbo-16k**. In **response_format** select **{"type":" text"}**.



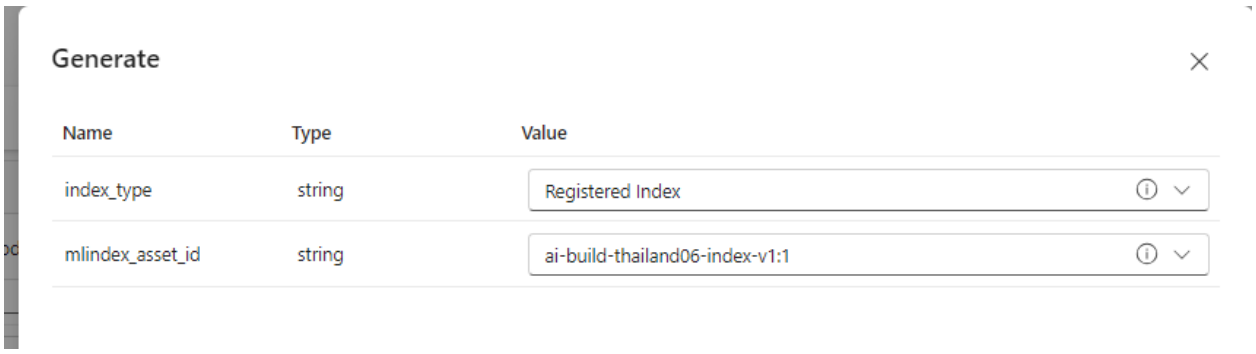
8. The **lookup** tab is for the retrieval system which uses Azure AI search service. Set up the connection to the created index by clicking on the **Value** box in the **mlindex_content.**



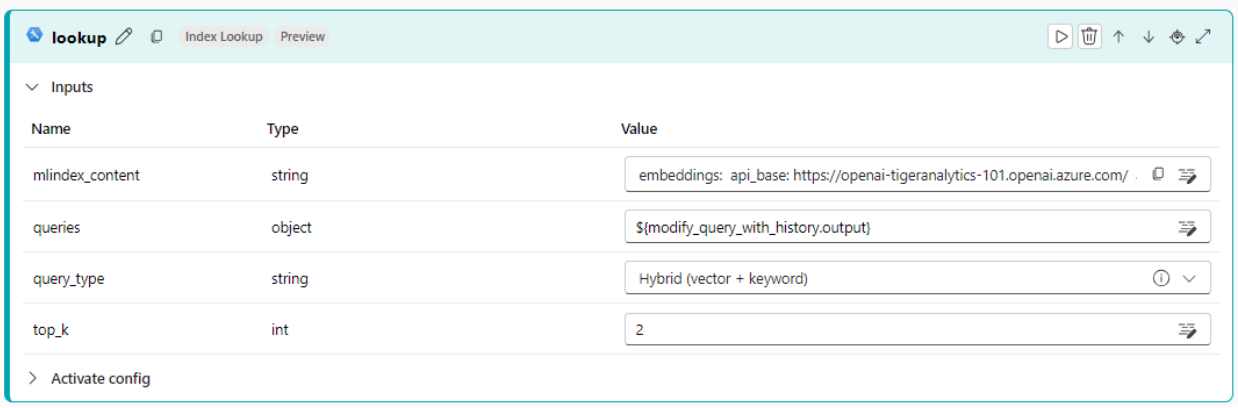
Under **index_type,** select **Registered Index**, the index already created in the playground setup.



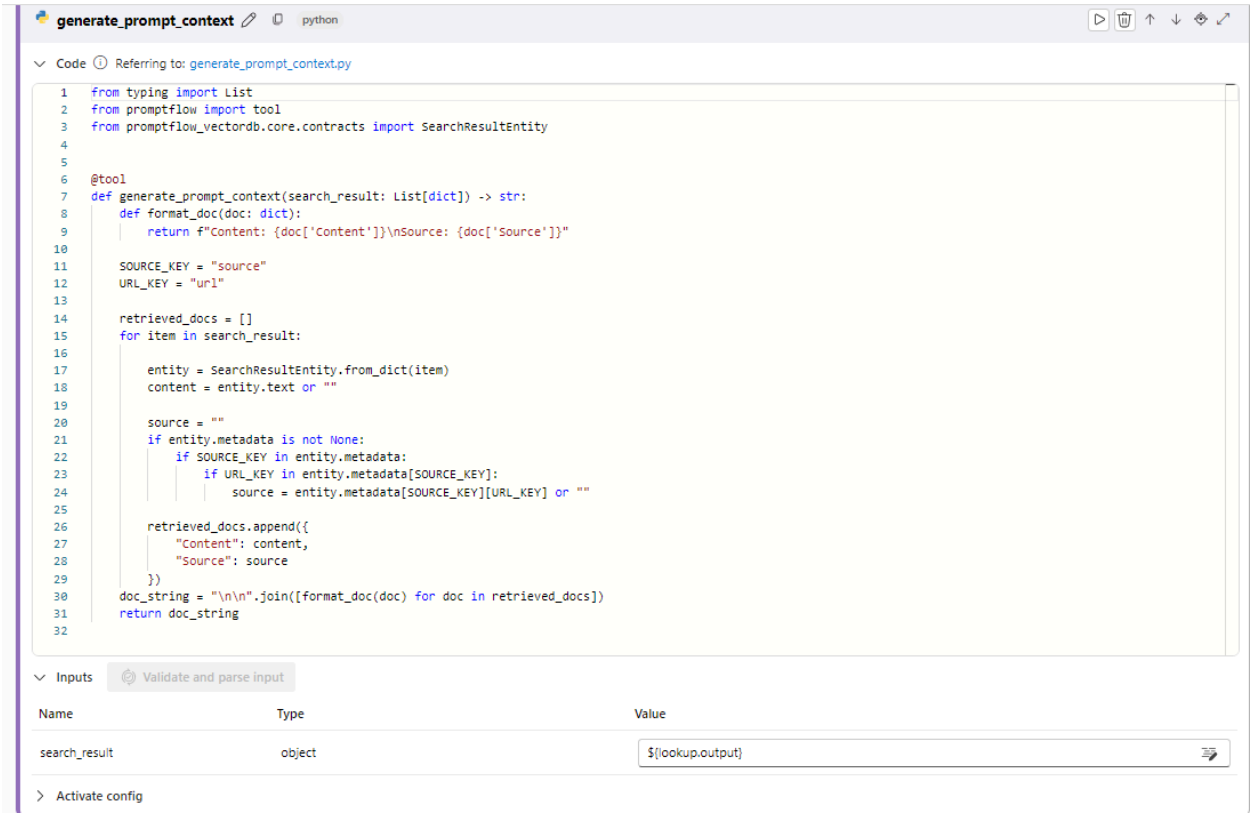
Select the index name under **mlindex_asset_id** that is already created. The create index name would be "**ai-build-**" + **user_id** + **version**. E.g. **ai-build-thailad06-index-v1**



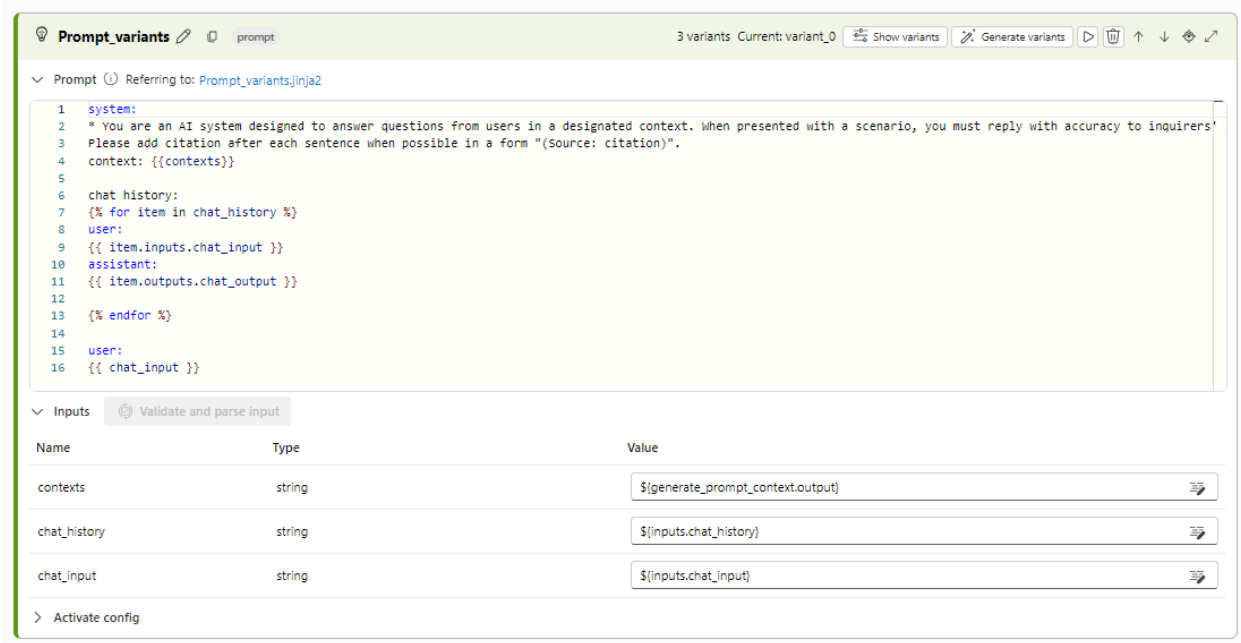
Select **Hybrid (vector + keyword)** under **query_type**. You can select other search types as well based on the use case. Specify **top_k** for the number of chunks you want to fetch as part of the search.



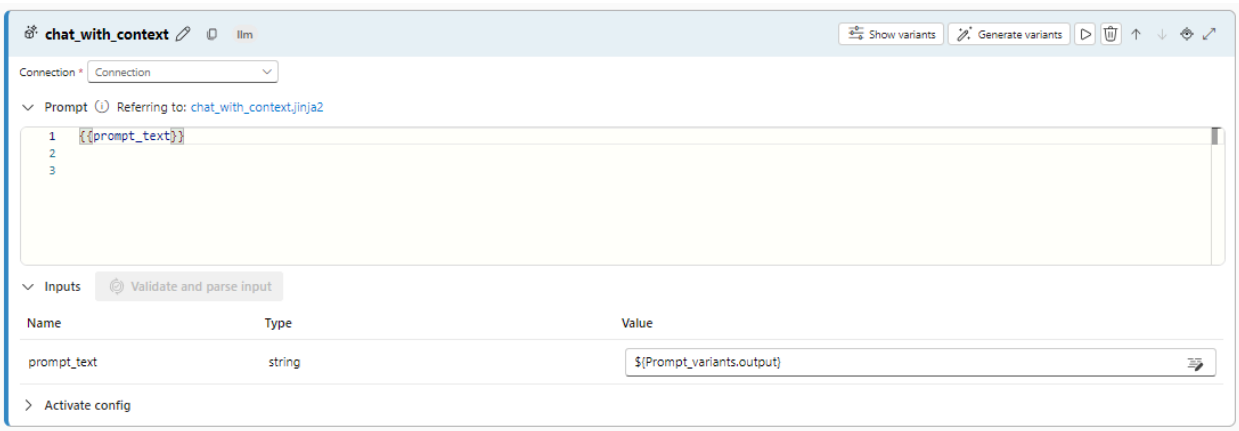
9. The **generate_prompt_context** tab is Python code that combines the context and source retrieved by Azure AI search.



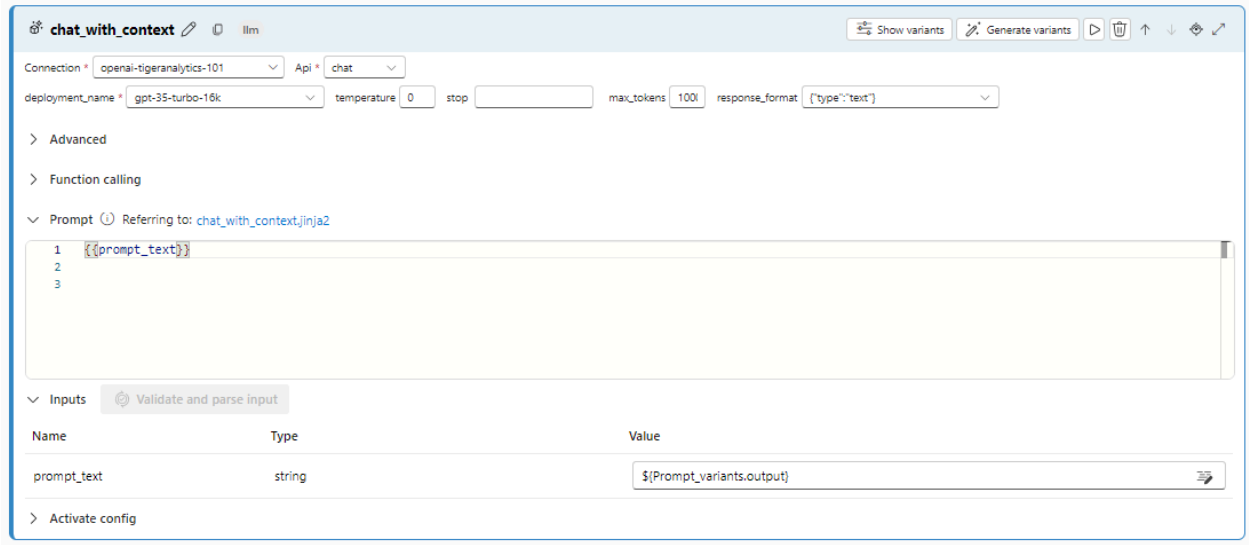
10. The **Prompt_variants** section combines the chat history, context, and system prompts to create a common prompt template to be fed to LLM for generating responses. We can tweak the template and perform prompt engineering as needed.



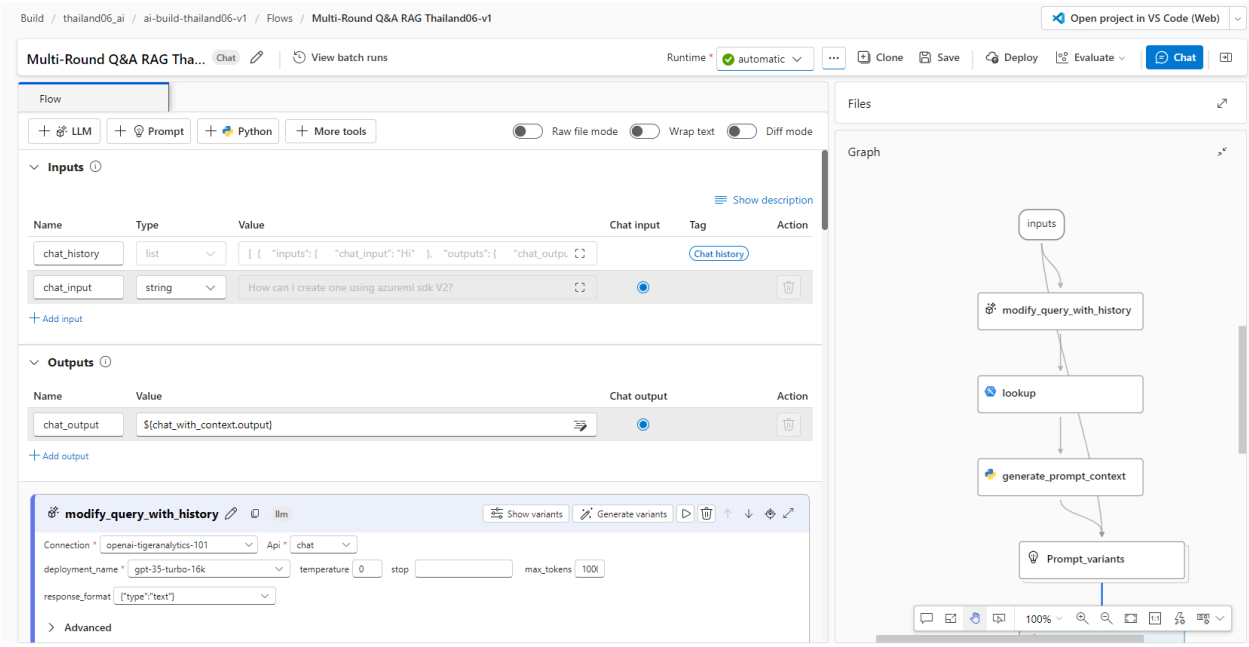
11. **Chat_with_context** takes the prompt from the above step and generates responses using LLM.



Set up the OpenAI connection in **chat_with_context**. 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"}**.



12. The prompt flow is ready to use and you can view the process flow from the **Graph** tab on the right section.



You can click the **Chat** option to open the chat window and interact with the application.