# Lab 07 : Getting Into the Flow

Up until now, you have used a web service (ChatServiceWebAPI) that utilizes an instance of the ChatService singleton to orchestrate calls to Azure OpenAI and Azure Cognitive Search by using Semantic Kernel. This effectively provides the "smarts" to your AI assistant. This is not the only way that you could build these smarts.

In this lab, you will use Azure ML Prompt Flow to replace portions of the ChatService singleton.

**Objective**.

Create an Azure Machine Learning Prompt Flow that re-creates the core steps of the ChatService, which are:

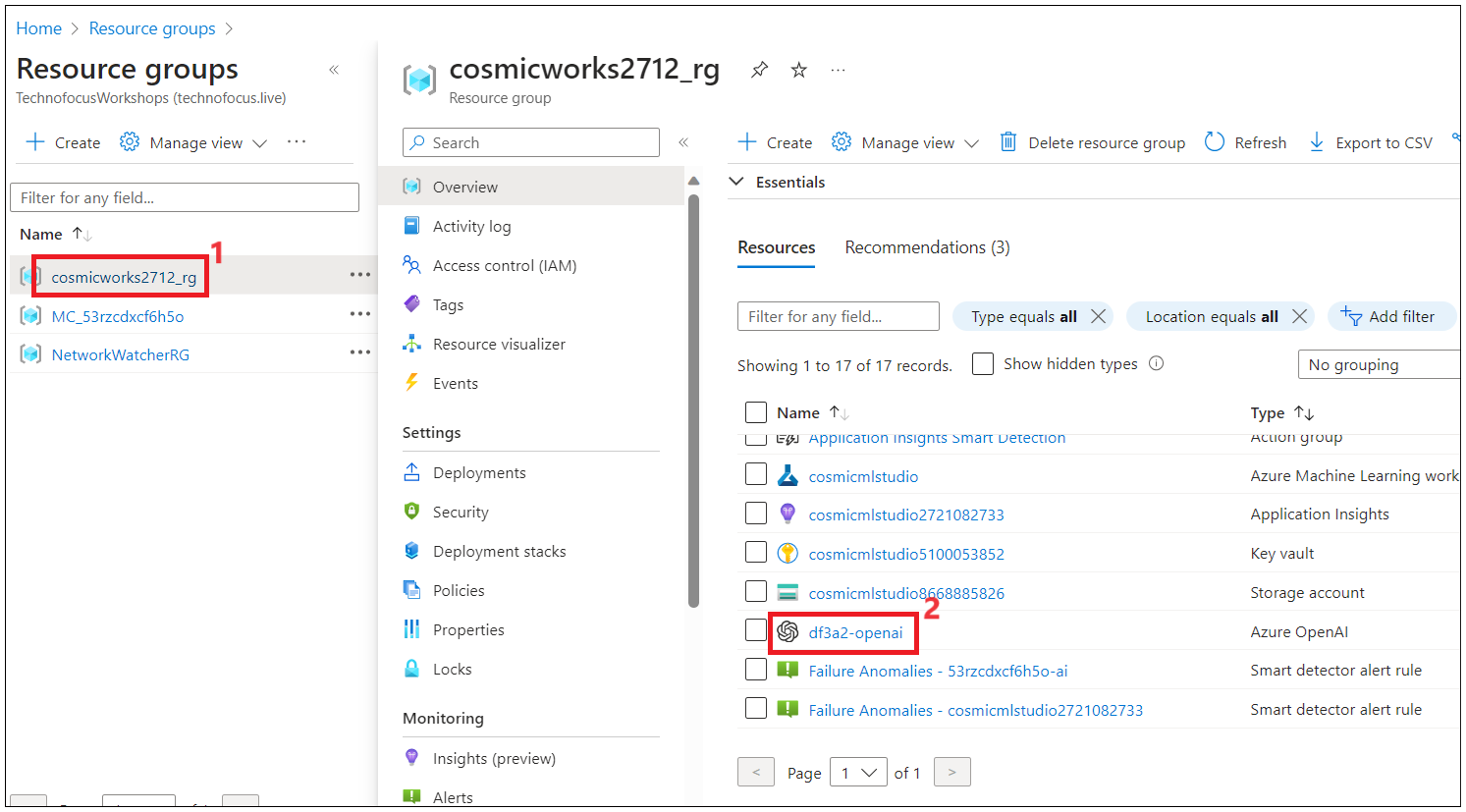
* + Get user query vector embedding
  + Search for context data
  + Request the completion
  + Store and return the result

**IMPORTANT:** Create and develop your own prompt flow and explore. We recommend the use of the **Chat flow** template or the **Bring Your Own Data QnA** starter from the gallery.

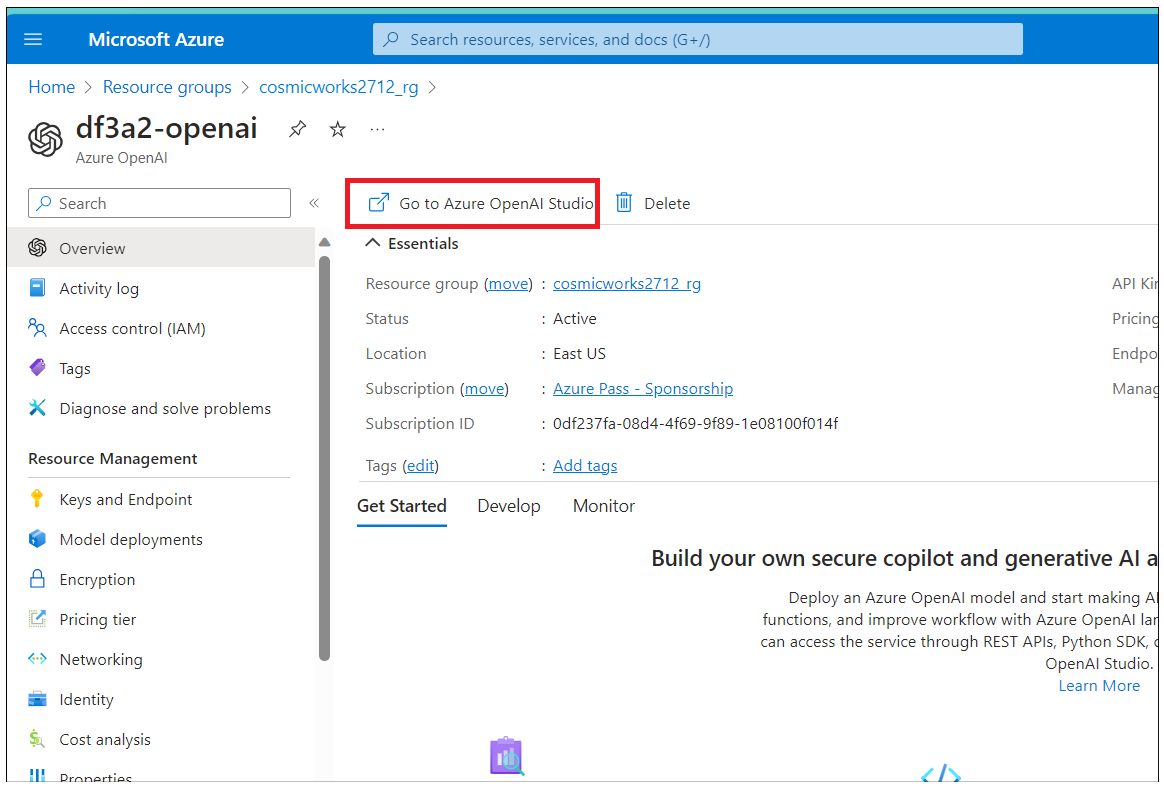
Some tasks are elaborated to get started with. You can still use your code and ideas to build the flow.

## **Task 1: Get OpenAI configuration values**

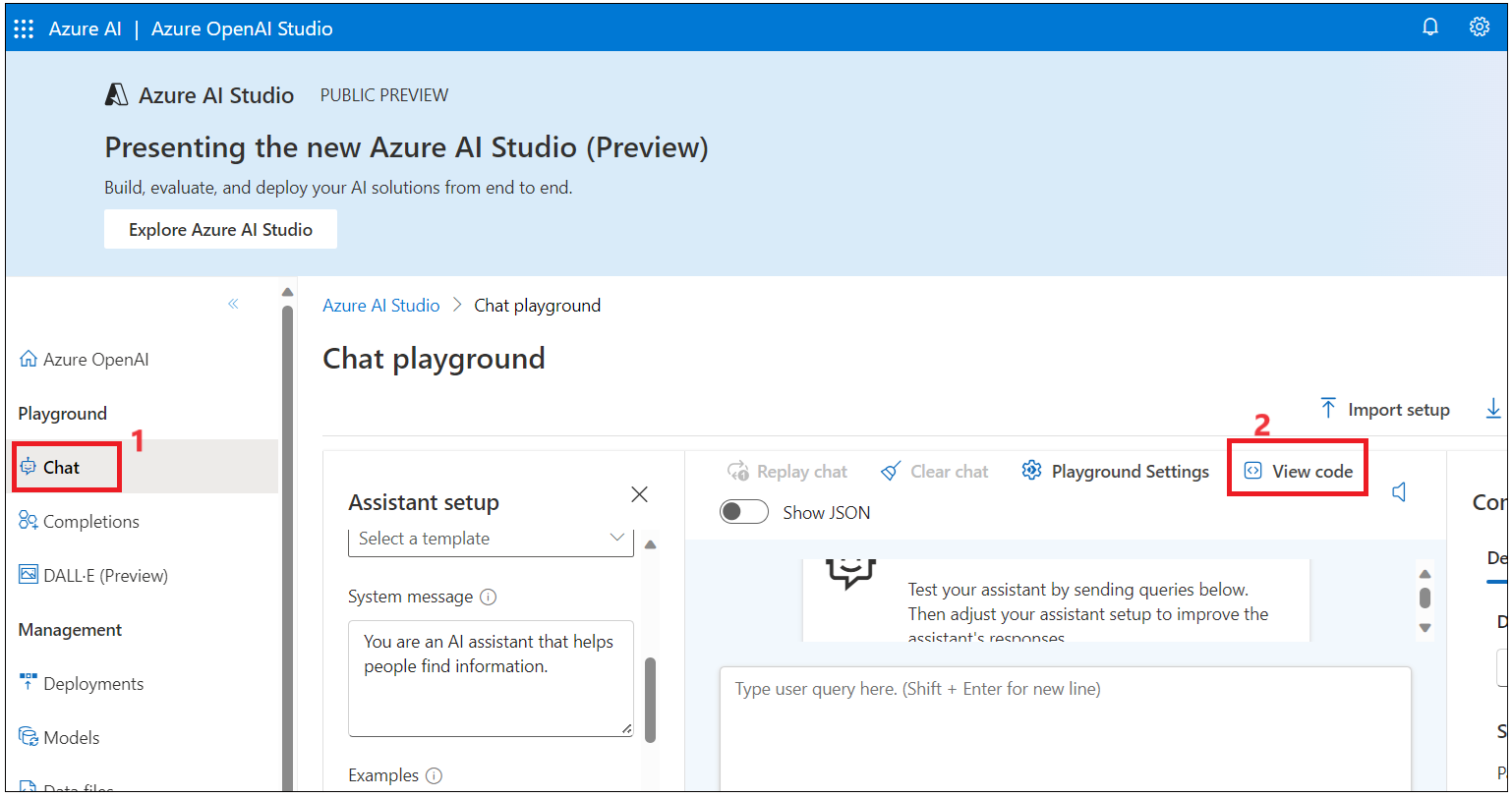
1. Switch back to the **Azure portal**, click on your resource group, and then click on the **Azure Open AI** resource name.

****

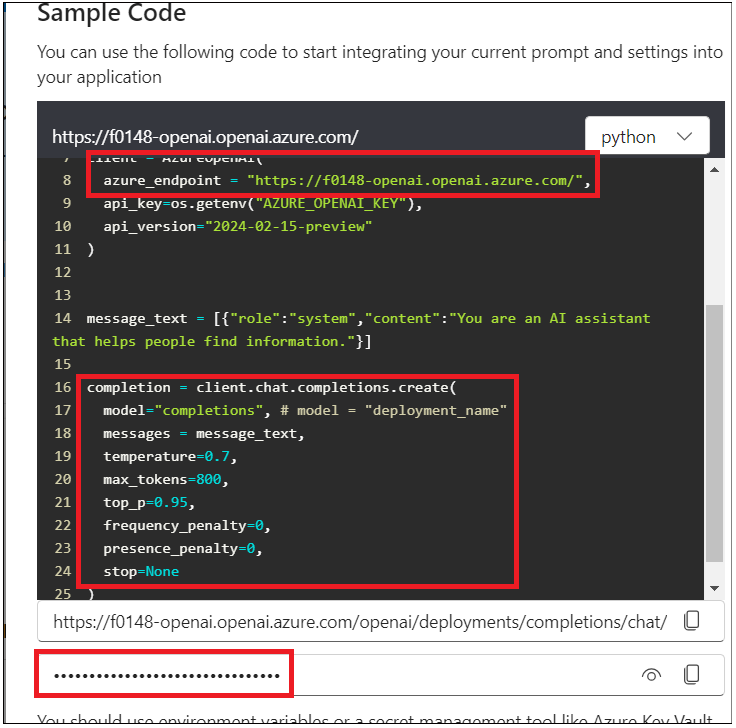
1. Click on **Go to Azure Open AI Studio.**

****

1. Click on **Chat** under **Playground** from the left navigation menu and then click on **View code** on the **Chat playground** page.

****

1. Copy the endpoint,key ,temperature ,top\_p values and save them in Notepad as these details are required to create prompt flow in the next task.

****

## **Task 2: Create the workspace and set up Azure OpenAI connection**

1. Open a new tab in your browser, go to +++<https://ml.azure.com/+++> , and sign in with your Azure subscription.

A screenshot of a computer

Description automatically generated

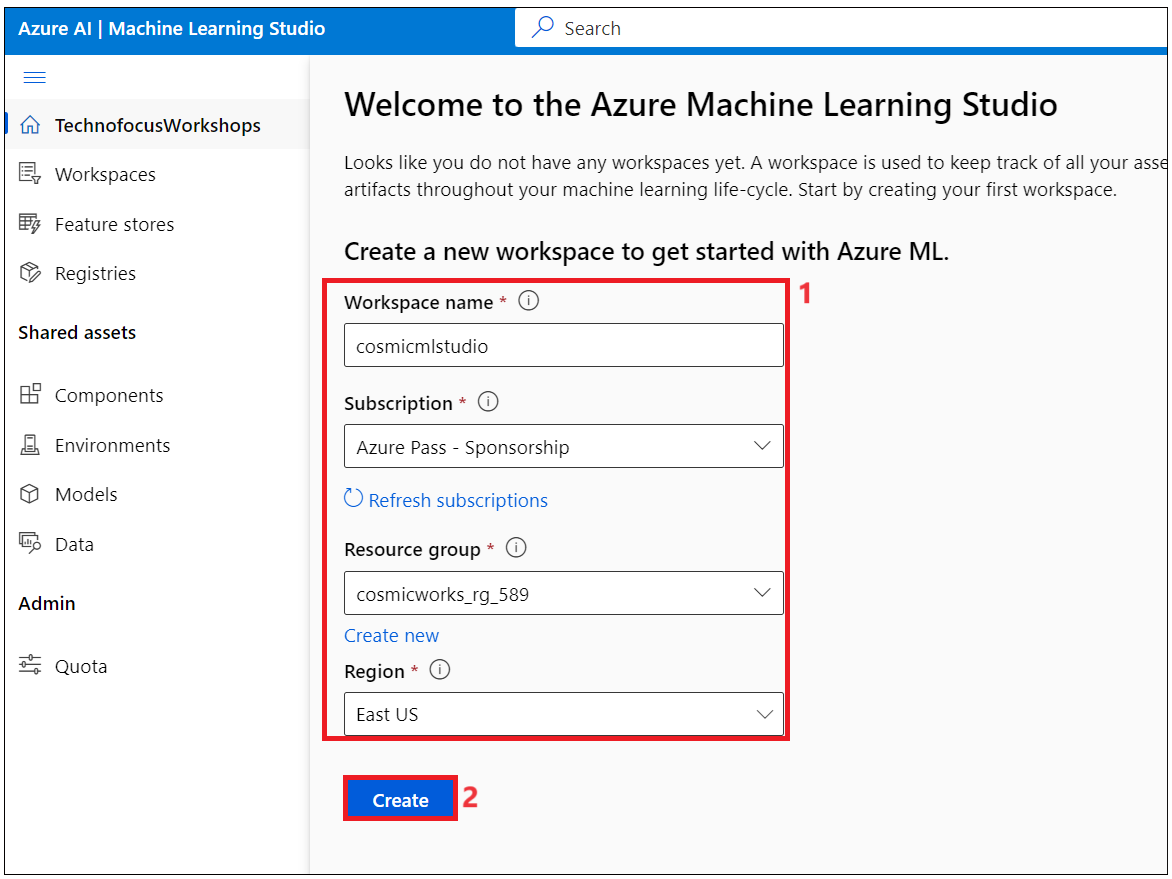
1. Provide the following information to configure your new workspace and then click on **Create**.

Workspace name: +++**cosmicmlstudio+++**

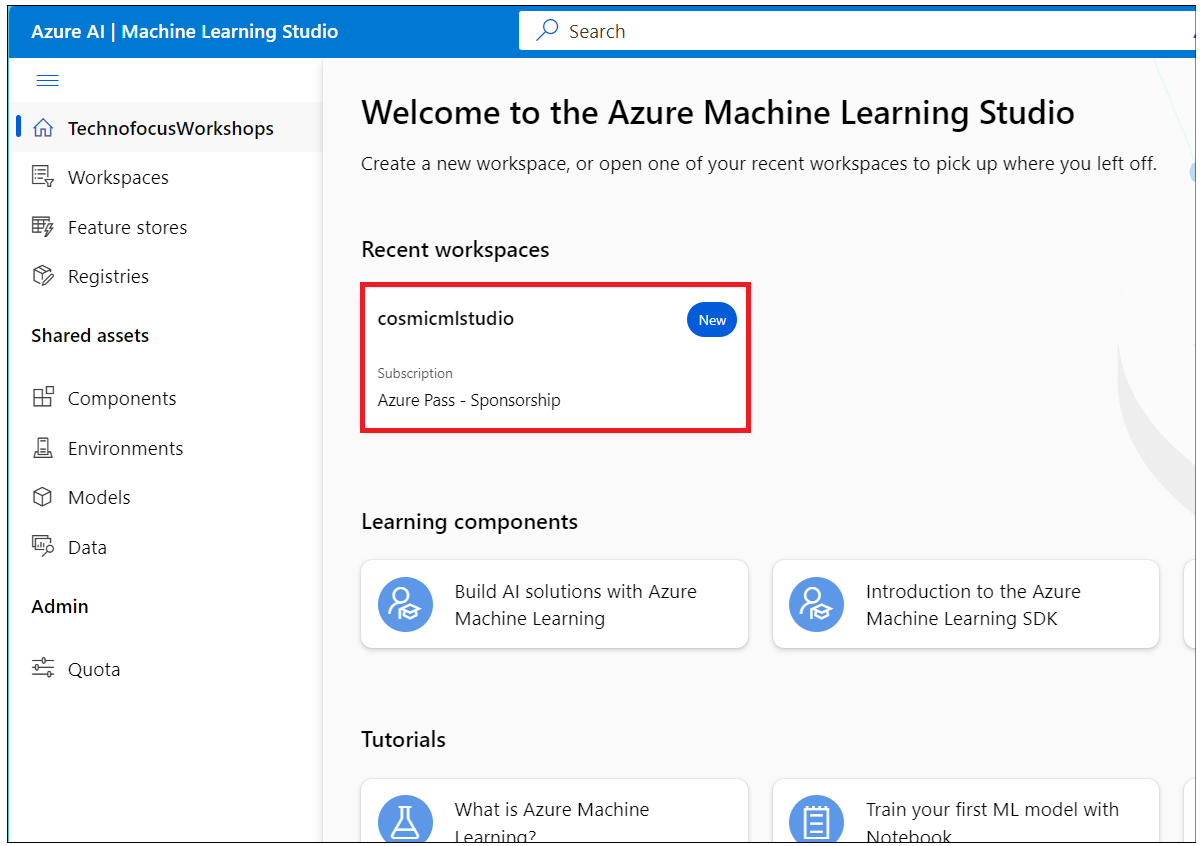
Subscription: **Your Azure subscription**

Resource group: Select Resource group – **cosmicworks\_rg\_XXXX**

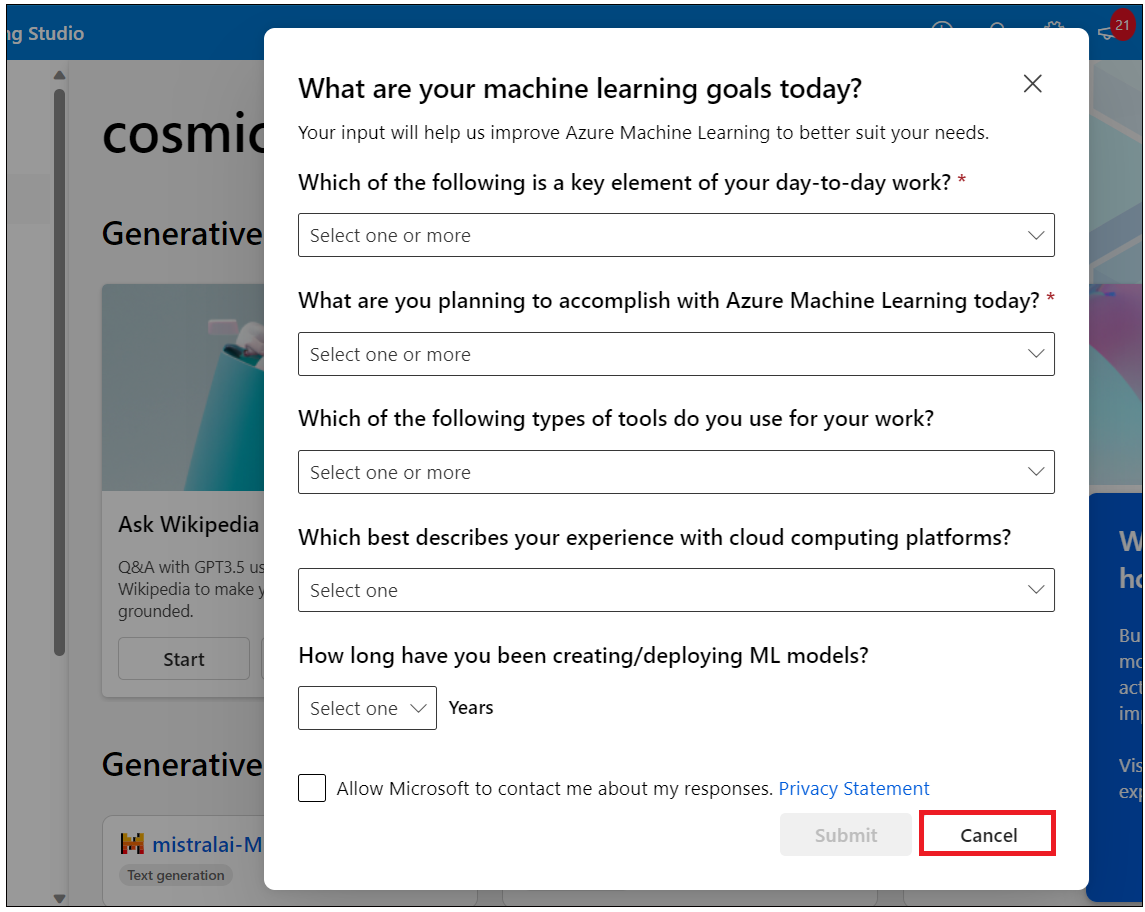
Region: East US



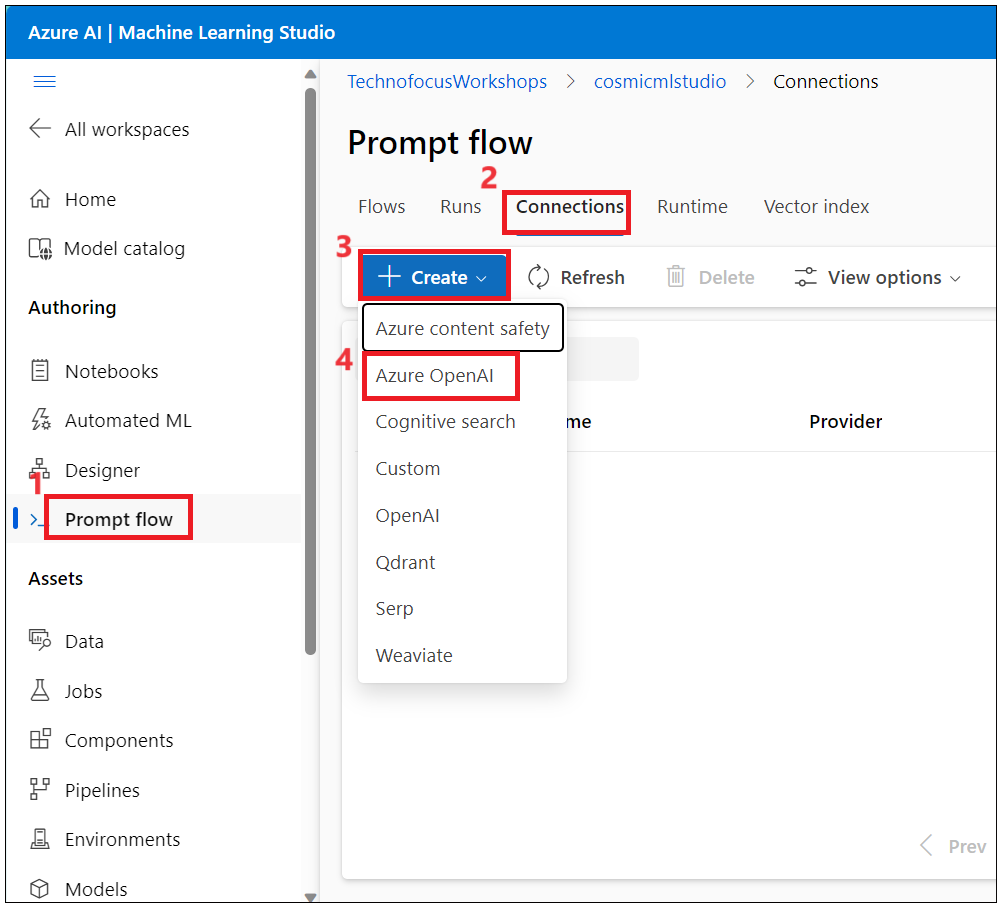
1. Wait for the workspace to be ready. Once it is ready then click on it.



1. Cancel the Feedback form.



1. Navigate to the **Prompt flow** under the **Authoring** section on the left menu. Click on **Connection** -> **Create** and select **Azure** **OpenAI**.



1. The **Add Azure OpenAI connection** panel will appear on the right-hand side. Enter the below details and then click on **Save**.

**Name: +++CosmicOpenAIconneciton+++**

**Provider: Azure OpenAI**

**Subscription id: Azure Pass – Sponsorship.**

**Azure OpenAI Account Names: your OpenAI account name**

**API key:** <Choose a key from 'Keys and endpoint' in your Azure OpenAI instance in the Portal> or get it from Sample code in Azure OpenAI studio (Task 1 -last step)

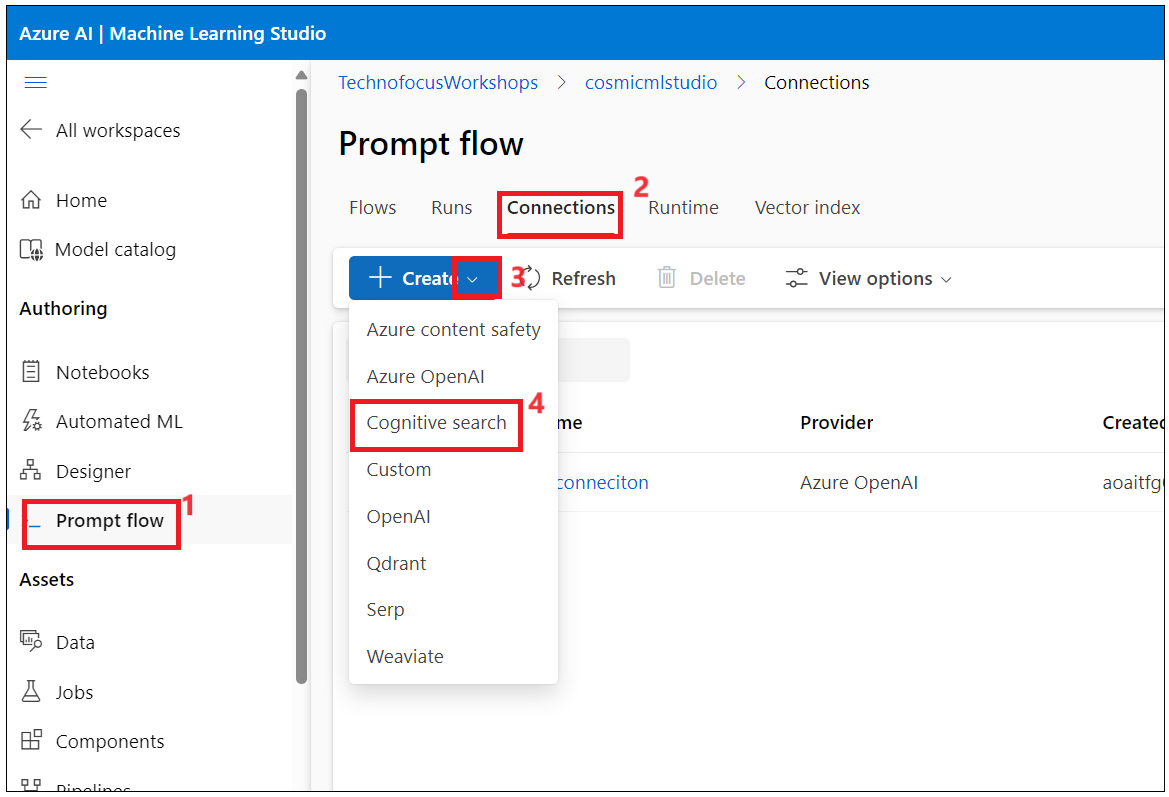
**API base:** endpoint

A screenshot of a computer

Description automatically generated

## **Task 3: Create Azure Cognitive Search Connection.**

1. Navigate to the **Prompt flow** under the **Authoring** section on the left menu. Click on **Connection** -> **Create** and select **Cognitive Search**.



1. The **Add Cognitive search connection** panel will appear on the right-hand side. Enter the below details and then click on **Save**.

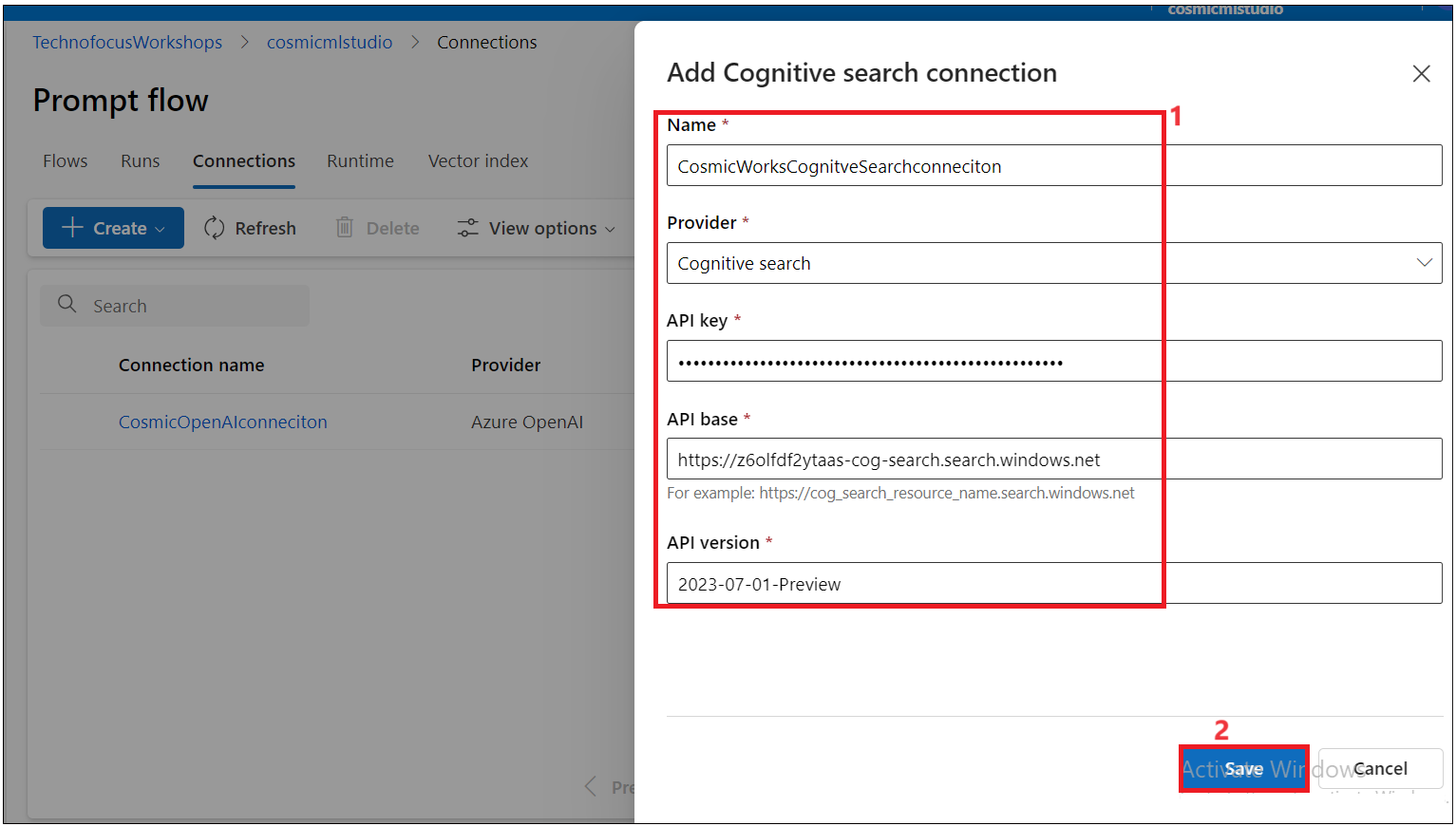
**Name: +++CosmicWorksCognitveSearchconneciton+++**

**Provider: Cognitive Search**

**API key:** Azure Cognitive Search service key

**API base:** Azure Cognitive Search service endpoint

**API version: the values copied in the previous task (Azure OpenAI studio-> Chat->View Code)**

****

## **Task 4: Create Azure Cosmos DB Connection.**

1. Click on **Create** and select the **Custom** Connection option as shown in the below image.

A screenshot of a computer

Description automatically generated

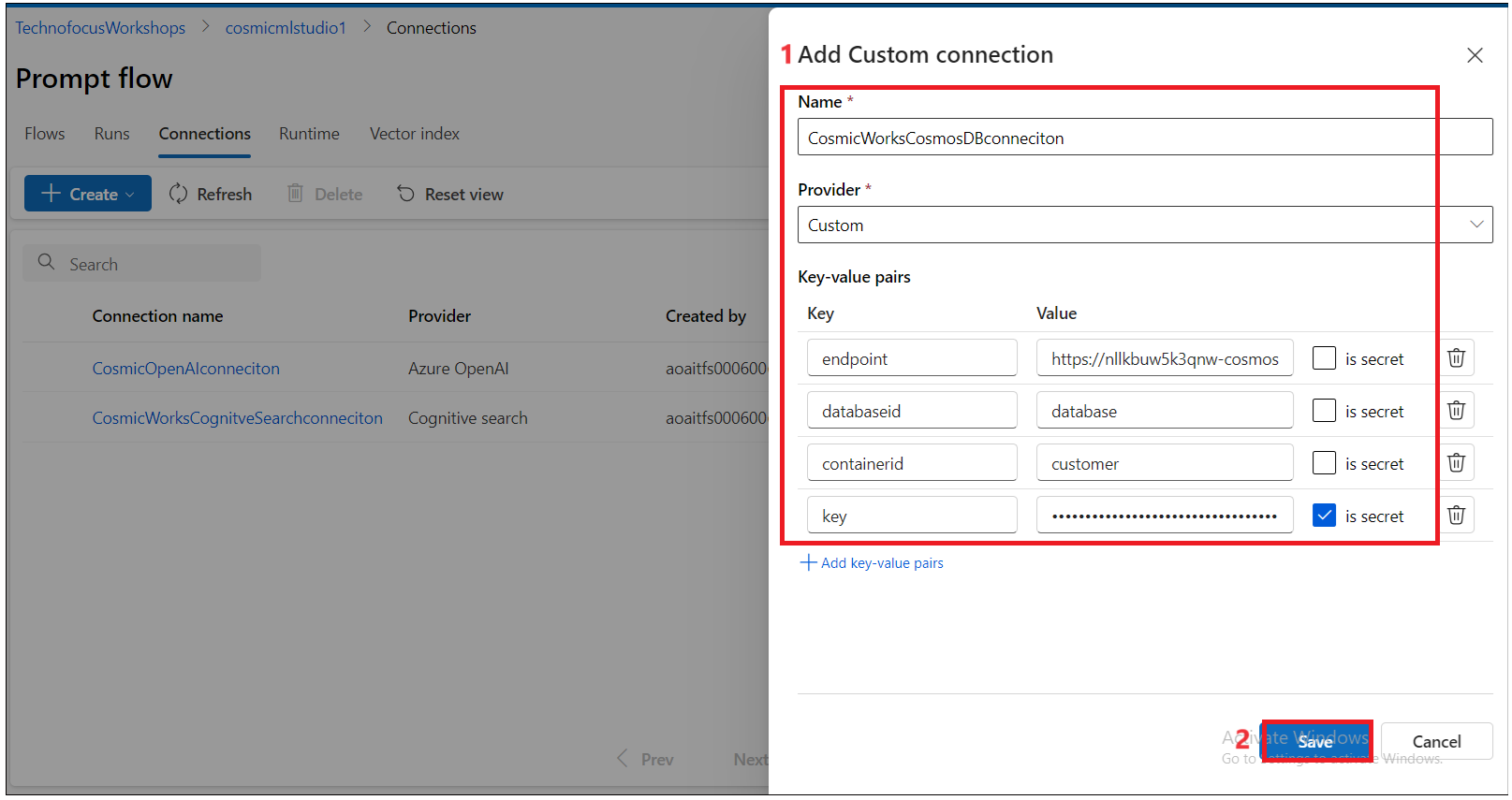
1. The **Add Azure OpenAI connection** panel will appear on the right-hand side. Enter the below details and then click on **Save**.

**Name: ++CosmicWorksCosmosDBconneciton++**

**Provider: Custom**

**Enter Key value pair as per below table**

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Is secret** |
| **+++endpoint+++** | Your Cosmosdb endpoint | No |
| **+++databaseid+++** | **+++database+++** | No |
| **+++containerid+++** | **+++customer+++** | No |
| **+++key+++** | Your primary key | Yes |

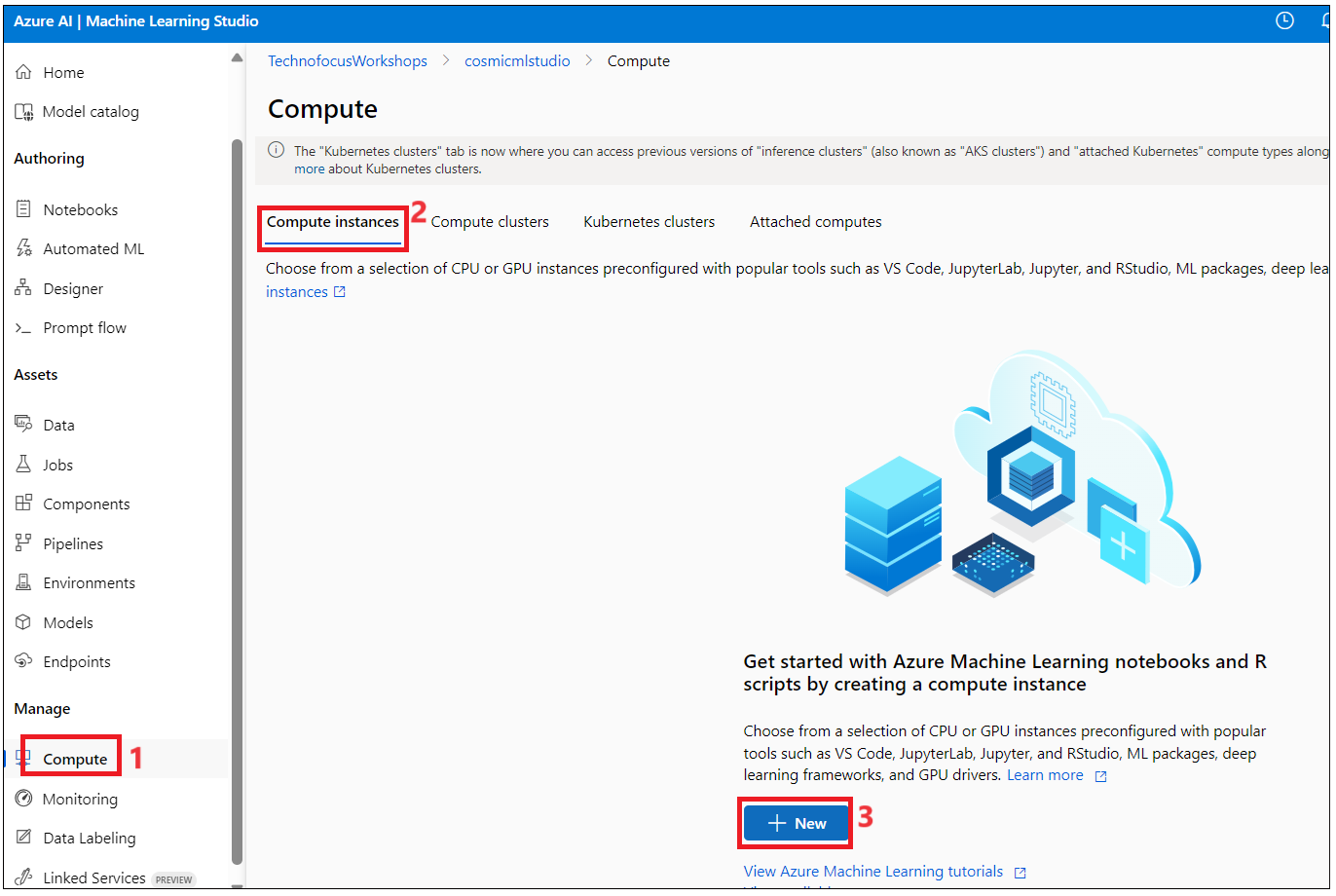


A screenshot of a computer

Description automatically generated

# Task 5 : Create a compute that is needed for prompt flow

1. Click on **Compute** under **Manage** from the left navigation menu. Click on **+New**.

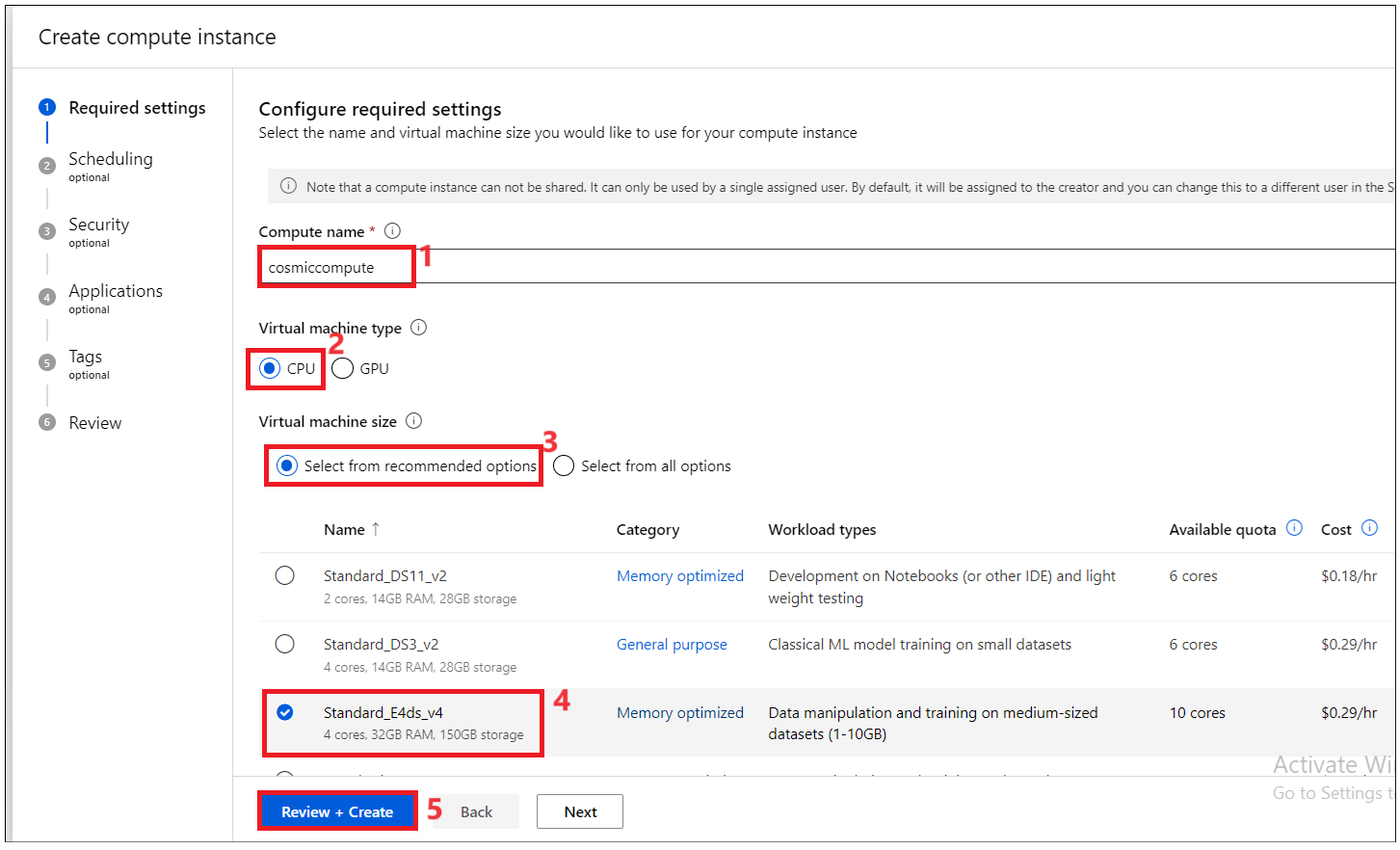


1. Enter the values below and then click on **Review + Create**.

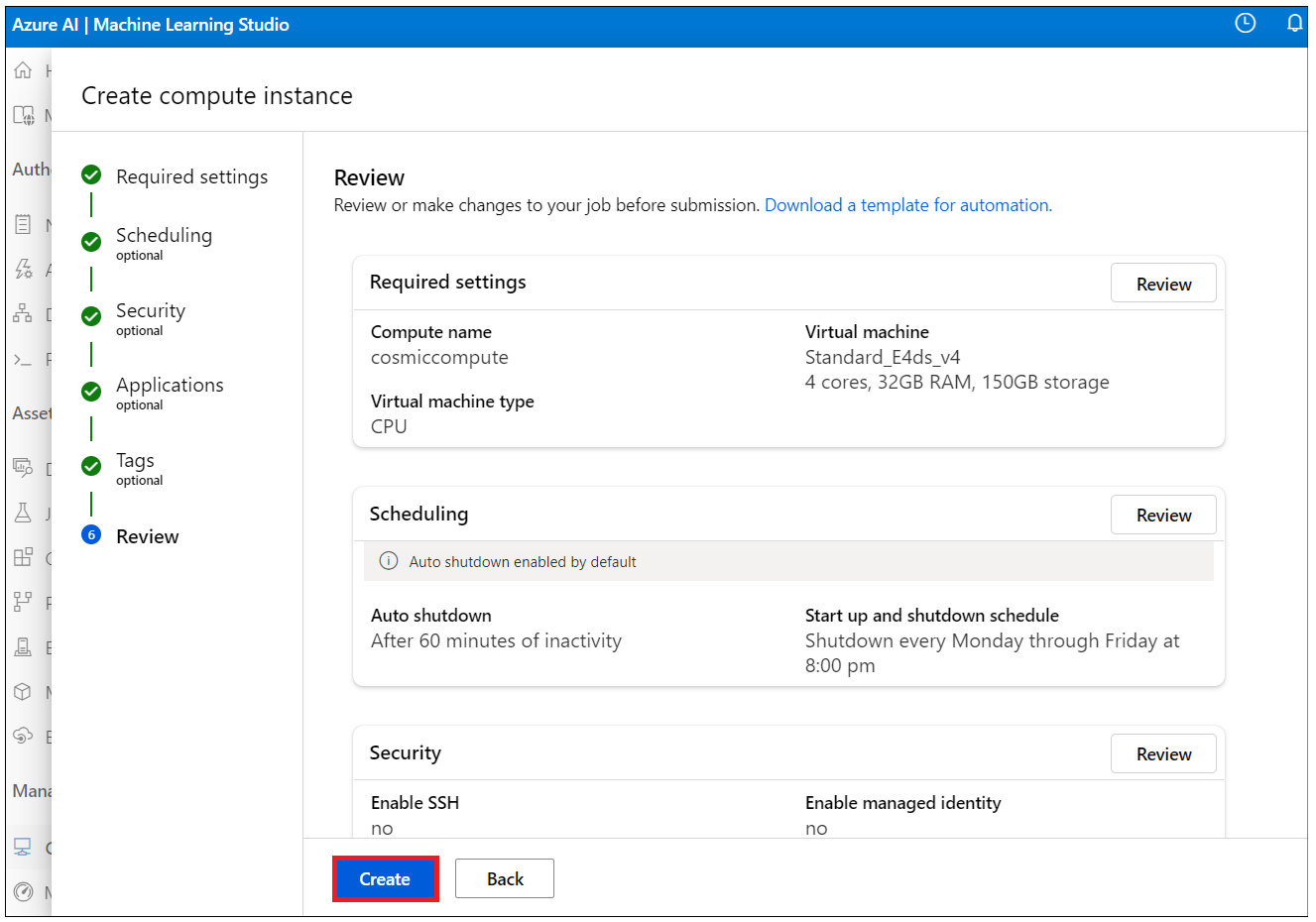
Compute Name : ++**cosmiccomputeXXX**++ (XXX can be a unique number)

Virtual machine type: **CPU**

Virtual machine size - **Standard\_E4ds\_v44 cores, 32GB RAM, 150GB storage**



1. In the **Review** pane, review the details that you’ve entered, and click on the **Create button.**



1. Wait for the Compute creation. It will take 15 -30 minutes to complete.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Task 5 : Create Runtime for the Prmopt flow

Make sure your Cosmicomute is up and Running

1. Click on **Prompt flow** under **Authoring** from the left navigation menu, and click on **Runtime -> Create.** Enter the below values and then clickon **Create.**

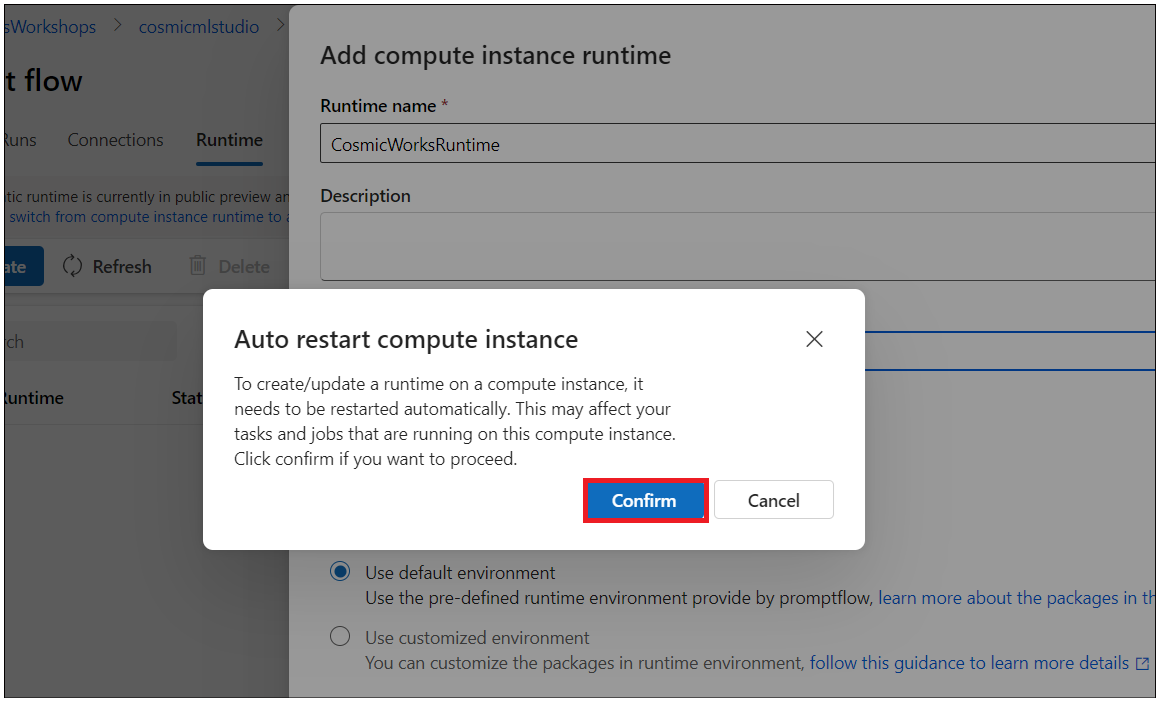
**Run time name: ++CosmicWorksRuntime++**

Select Azure ML compute instance: Cosmicomute

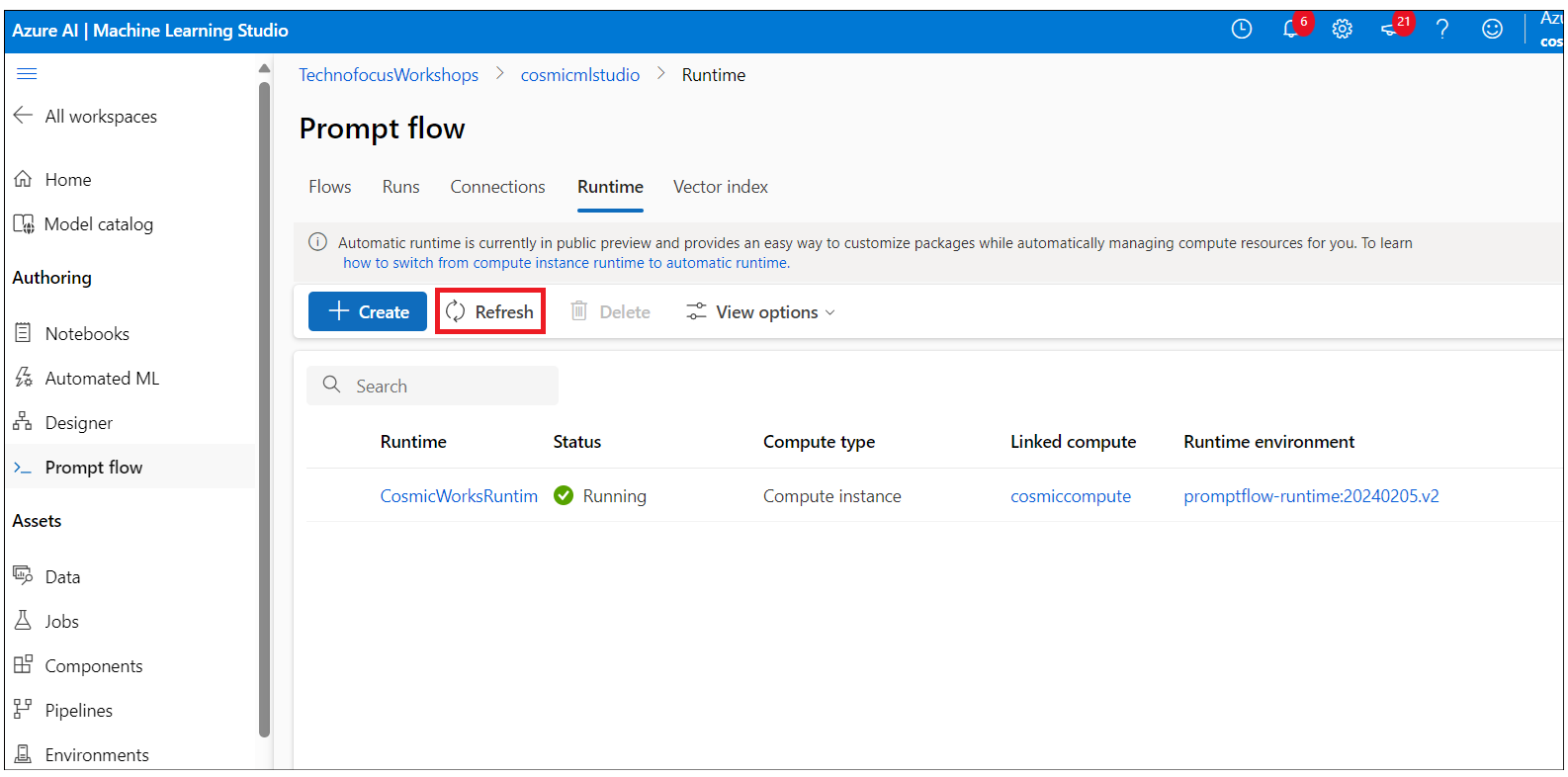
**A screenshot of a computer

Description automatically generated**

1. Click on **Confirm** to **Auto restart the compute instance**.



1. Click on Refresh until your Runtime is up and running. It takes < 1 min.



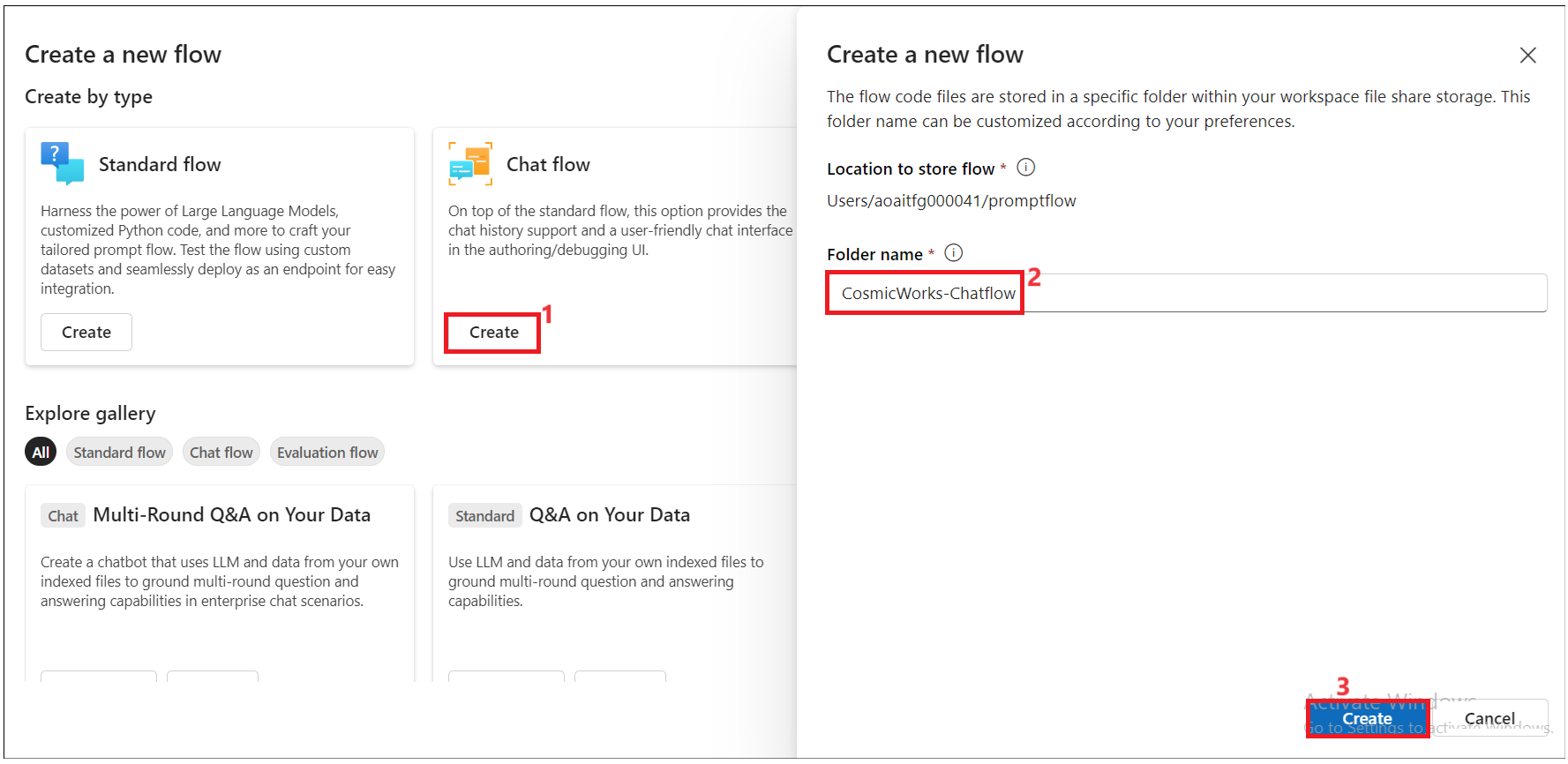
### **Task 6 : Develop the flow using different tools**

1. Click on **Prompt flow** under **Authoring** from the left navigation menu, and click on **Flows -> Create.**

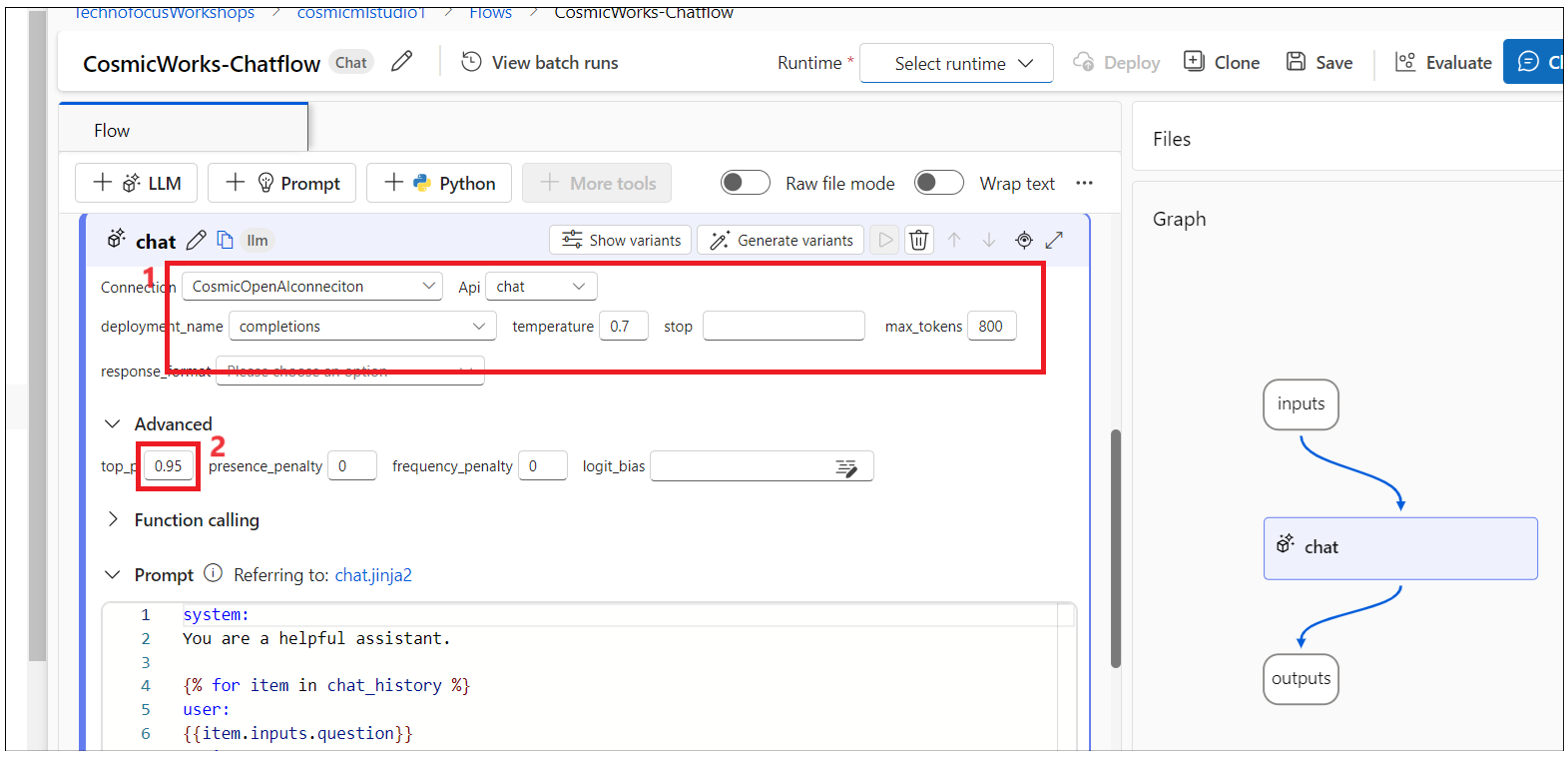
A screenshot of a computer

Description automatically generated

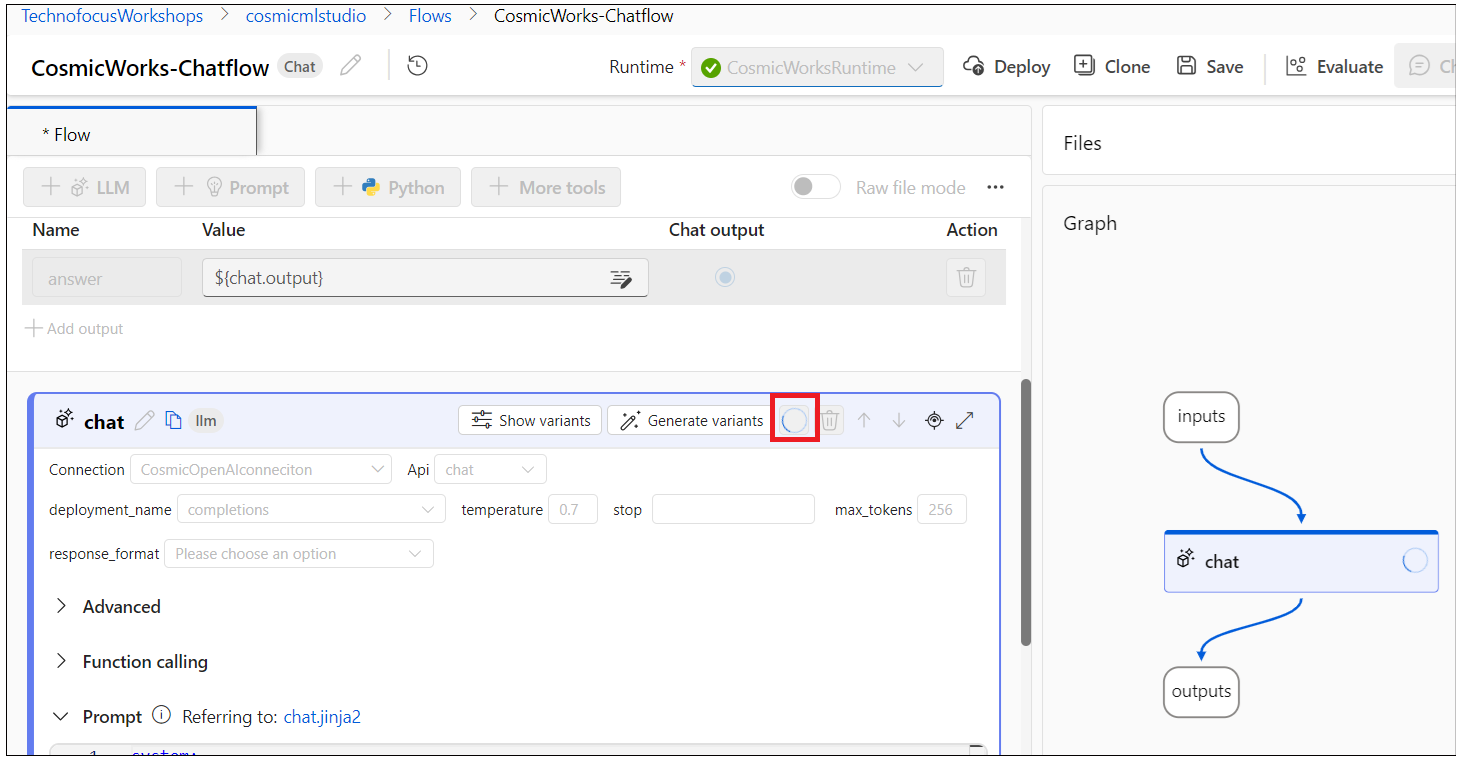
1. On the **Create a new flow** window, click on Create under the **Chat flow** tile . Enter the Folder name ++**CosmicWorks-Chatflow++** and then click on **Create**.



1. Select connection ----re-write the step.

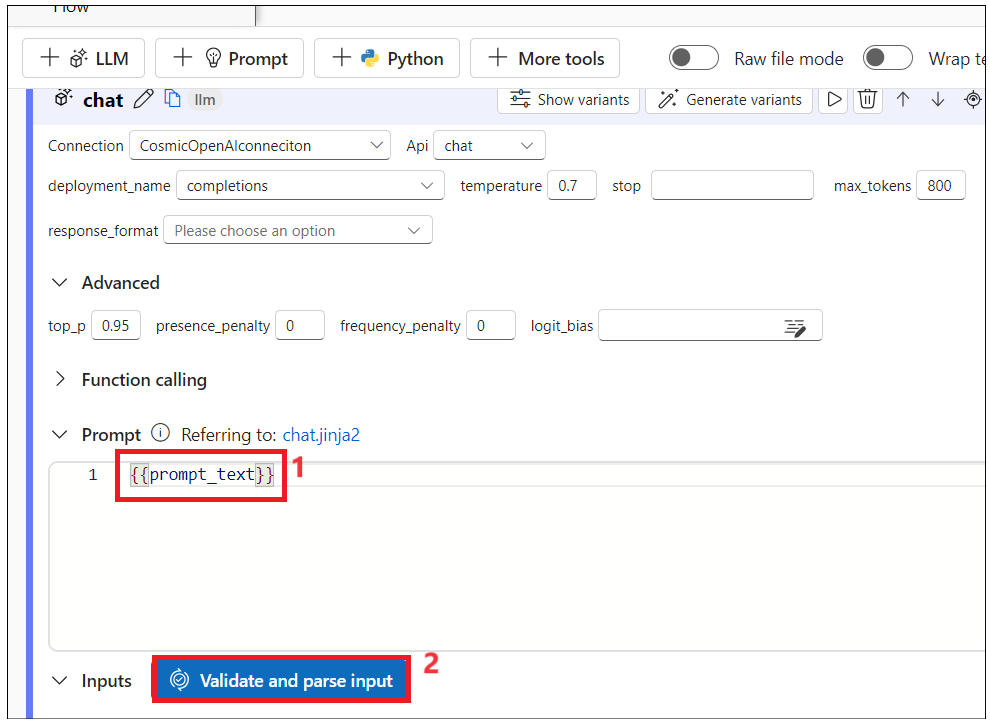


1. Click on Run next to the **Chat** node as shown in the below image.

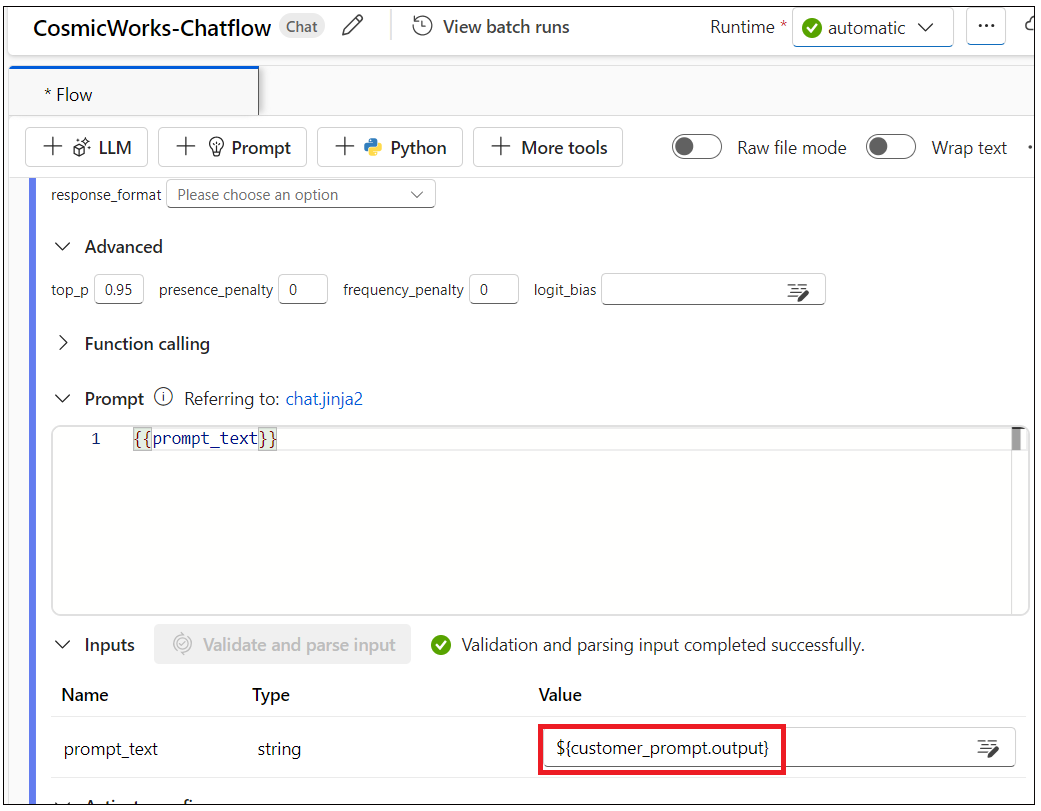


1. Enter the below code

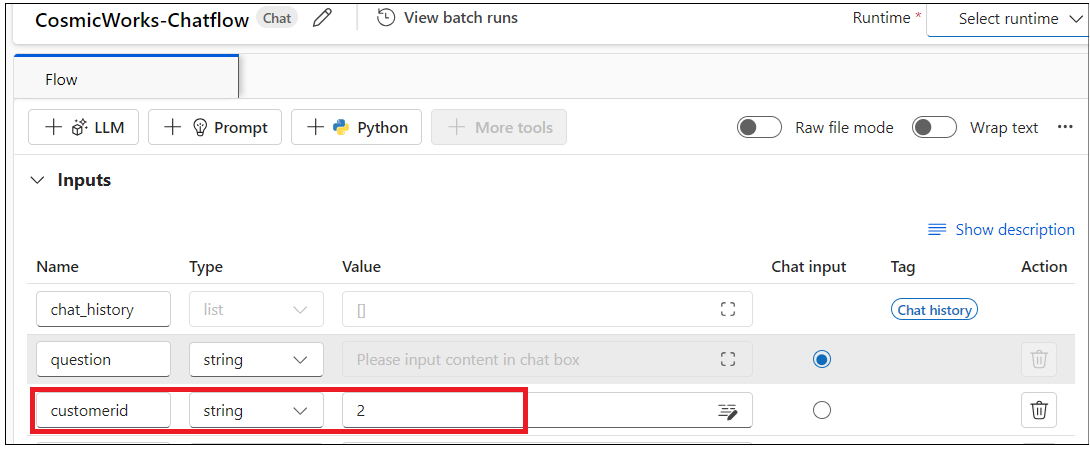
{{prompt\_text}}



1. Select prompt\_text as **${customer\_prompt.output}**

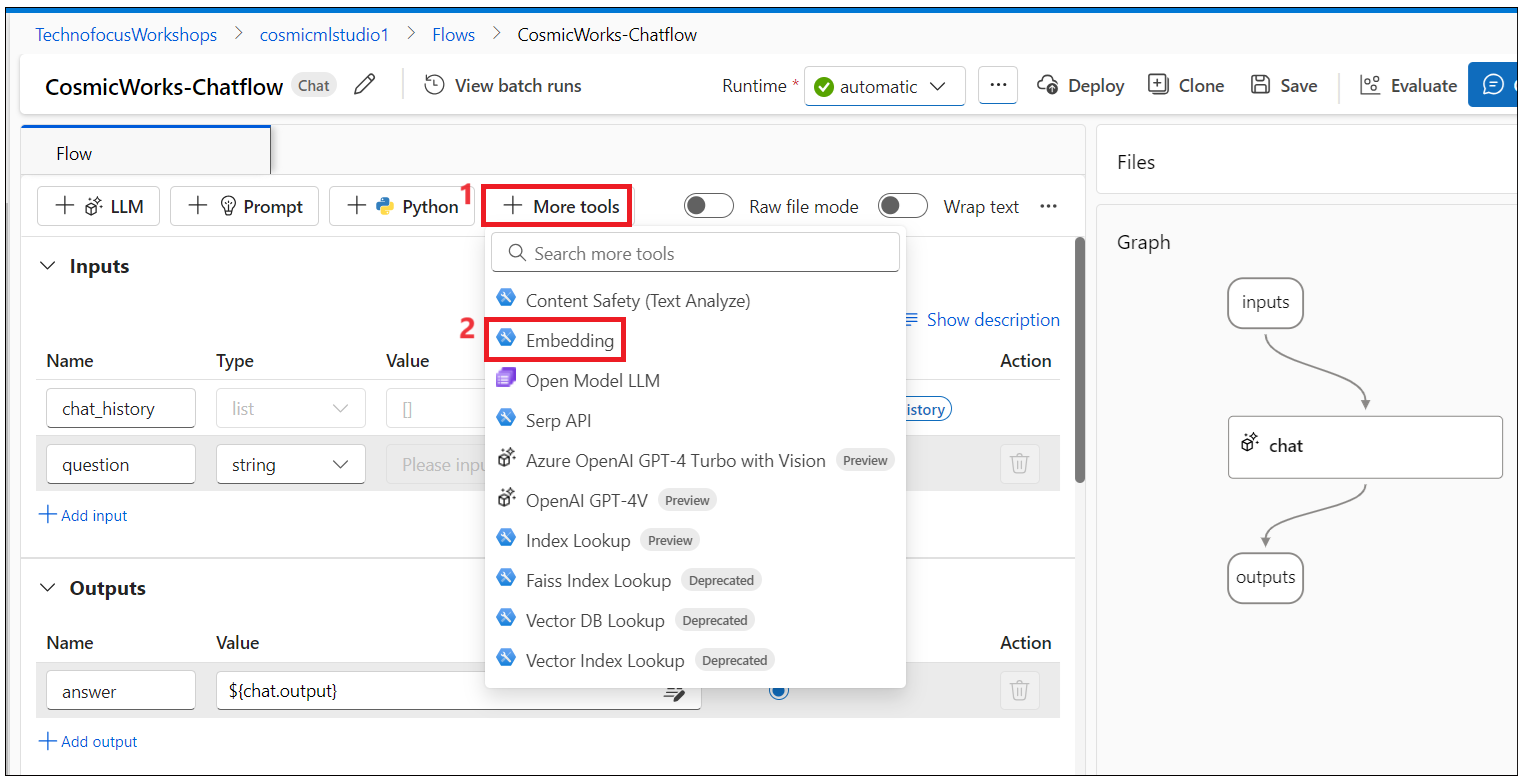


1. Scroll up and click on Add input and add +++customerId+++ as input string and value 2.

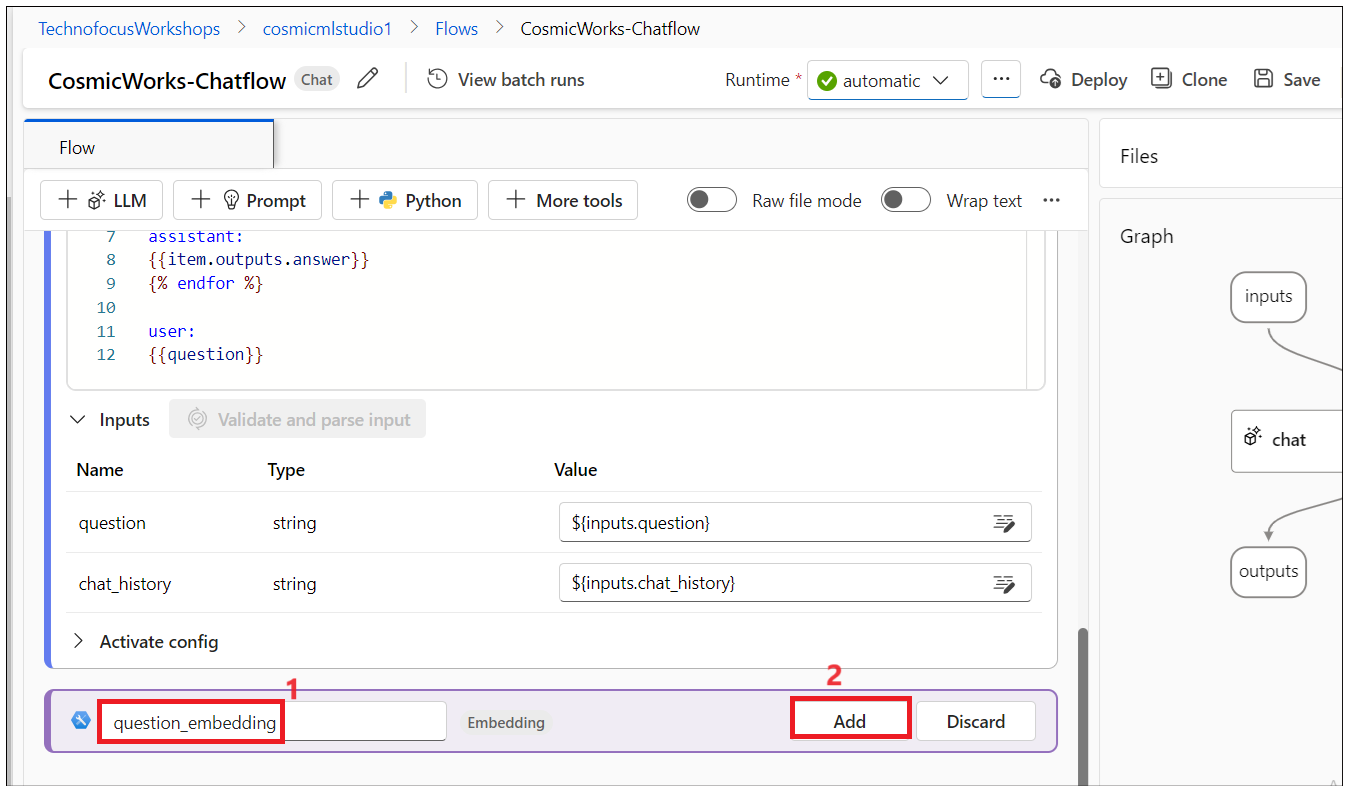


### **Task 7 : Add embedding tools for all interactions with Azure OpenAI**

1. Click on **More tools -> Embedding** shown in the below image



1. Enter name : +++**question\_embedding+++** and then click on **Add** as shown in the below image



1. Add the below code and then click on **Validate and parse input**

from typing import List

from promptflow import tool

from azure.search.documents import SearchClient

from azure.search.documents.models import Vector

from azure.core.credentials import AzureKeyCredential

from promptflow.connections import CognitiveSearchConnection

@tool

def retrieve\_documentation(question: str, index\_name: str, embedding: List[float], search: CognitiveSearchConnection) -> str:

  search\_client = SearchClient(endpoint=search.api\_base,

                                index\_name="vector-index",

                                credential=AzureKeyCredential(search.api\_key))

  vector = Vector(value=embedding, k=2, fields="embedding")

  results = search\_client.search(

    search\_text=question,

    top=2,

    search\_fields=["Id"],

    vectors=[vector],

  )

  docs = [{"id": doc["id"]}

          for doc in results]

  return docs

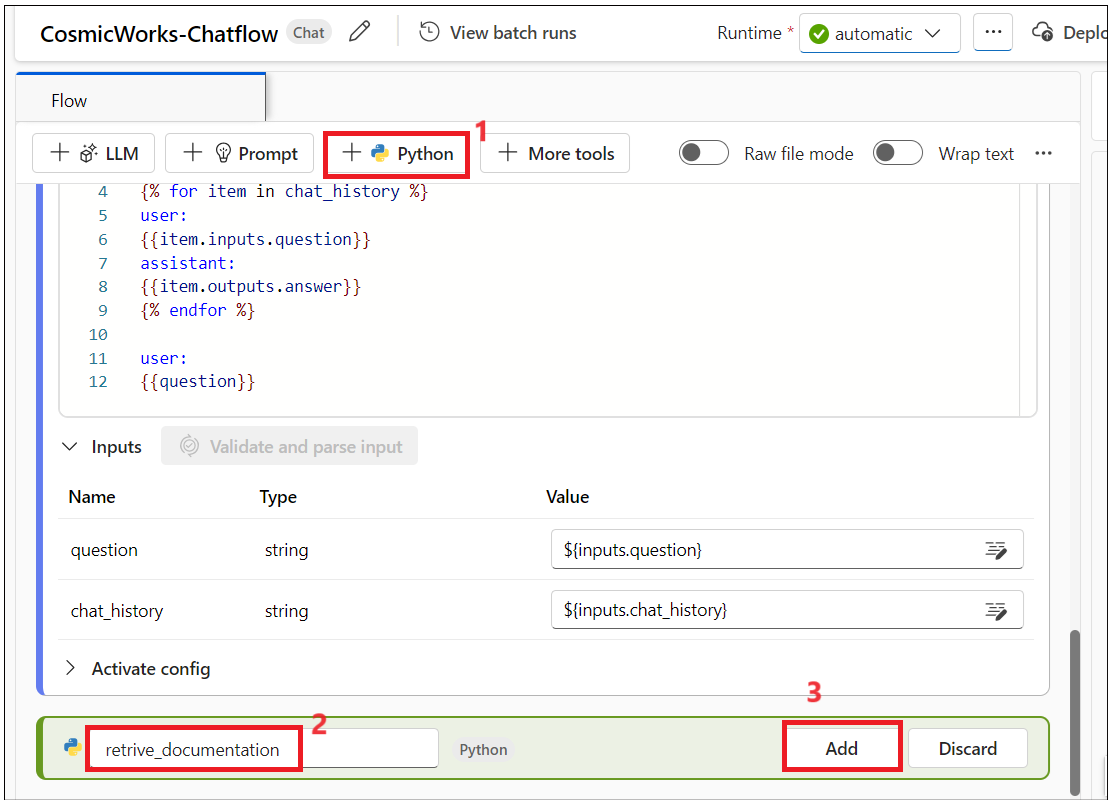
1. Select the below values and then click on
   1. Connection : **CosmicOpenAIconnection**
   2. Deployment\_name : **embeddings**
   3. Input : **$(inputs.question)**

**A screenshot of a computer

Description automatically generated**

### **Task 8 : Retrieve documentation from Azure Cognitive search**

1. Click on **Python,** enter the name +++**retrive\_documentation**+++ and then click on Add.



1. Enter the below code and click on Validate

from typing import List

from promptflow import tool

from promptflow.connections import CognitiveSearchConnection

@tool

def retrieve\_documentation(question: str, embedding: List[float], search: CognitiveSearchConnection):

  search\_client = SearchClient(endpoint=search.api\_base,

                                index\_name=vector-index,

                                credential=AzureKeyCredential(search.api\_key))

  vector=Vector(value=embedding, k=2, fields="embedding"),

  results = search\_client.search(

    search\_text=question,

    top=2,

    search\_fields=["Id"],

    vectors=[vector],

)

  docs = [{"id": doc["id"], "categoryName": doc["categoryName"], "sku": doc["sku"], "name": doc["name"]}

        for doc in results ]

A screenshot of a computer program

Description automatically generated

### **Task 9 : Homework tasks**

1. Click on the **prompt** tool, and create +++**customer\_prompt+++** to respond to questions only about products, customers, and account information. Validate ,select runtime and and run the flow.

A screenshot of a computer

Description automatically generated

1. Create a **customer\_lookup** to get the customer details and store the results back to Cosmos DB, consider using the Python Tool. Validate ,select runtime and run the flow

A screenshot of a computer

Description automatically generated

1. Deploy your flow to an online endpoint for integration with your application.
2. Develop a new class (named **AMLPromptFlowService**) that implements the **IRAGService** interface. This class will be responsible for calling the Azure ML Prompt Flow endpoint.
3. Update the dependency injection configuration in the Program class to use the new AMLPromptFlowService class instead of the SemanticKernelRAGService class.