Connect to Azure OpenAI in Python

Created by David Song, May 25, 2025

Contents

[Preface 1](#_Toc199078723)

[Create a Resource on Azure Portal 1](#_Toc199078724)

[Use Python to create a chat bot 4](#_Toc199078725)

[Install Python packages 4](#_Toc199078726)

[Create a file named: “.env” 4](#_Toc199078727)

[Source code 5](#_Toc199078728)

[Connect to Azure OpenAI 5](#_Toc199078729)

[Required package 5](#_Toc199078730)

[Source Code 5](#_Toc199078731)

[Generate two dataframe differences report from OpenAI 6](#_Toc199078732)

[Scouce code 6](#_Toc199078733)

[result 7](#_Toc199078734)

[References: 8](#_Toc199078735)

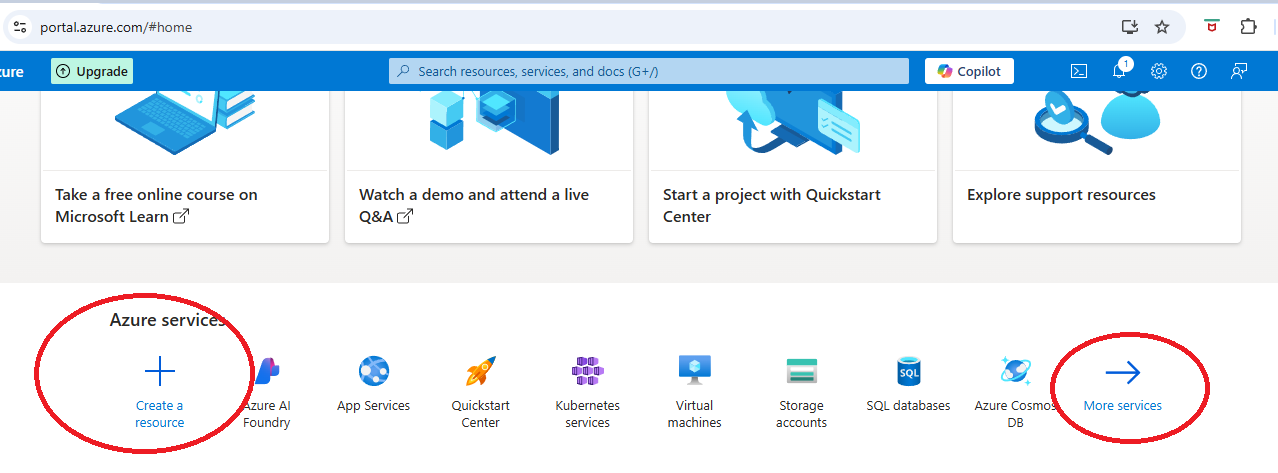
# Preface

This document shows how to use Python to connect to Azure OpenAI. Generally speaking, you will need to create a resource in Azure, get the end point and api keys, and then use the end point and api leys to connect to Azure Open Ai. It’s simple like that.

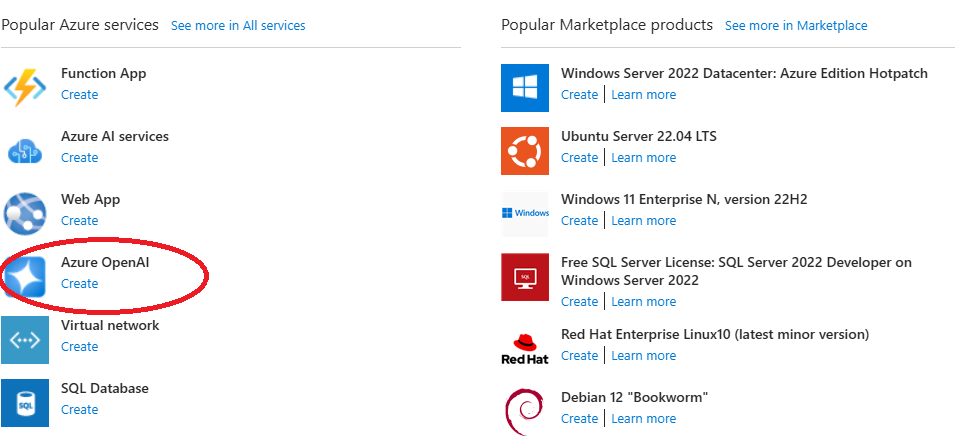
There are some differences to use Azure OpenAI and use OpenAI directly. We will only focus on Azure OpenAI at this moment.

# Create a Resource on Azure Portal

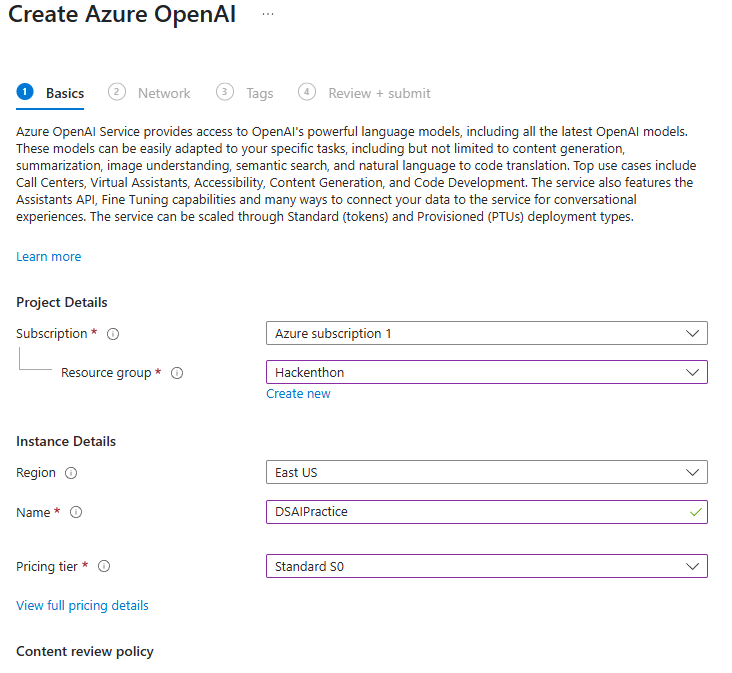
1. Login to Azure portal <https://portal.azure.com/#home> (Create a Azure account if you don’t have one).
2. Click on Create a resource or More Services:



1. Click on Azure OpenAI

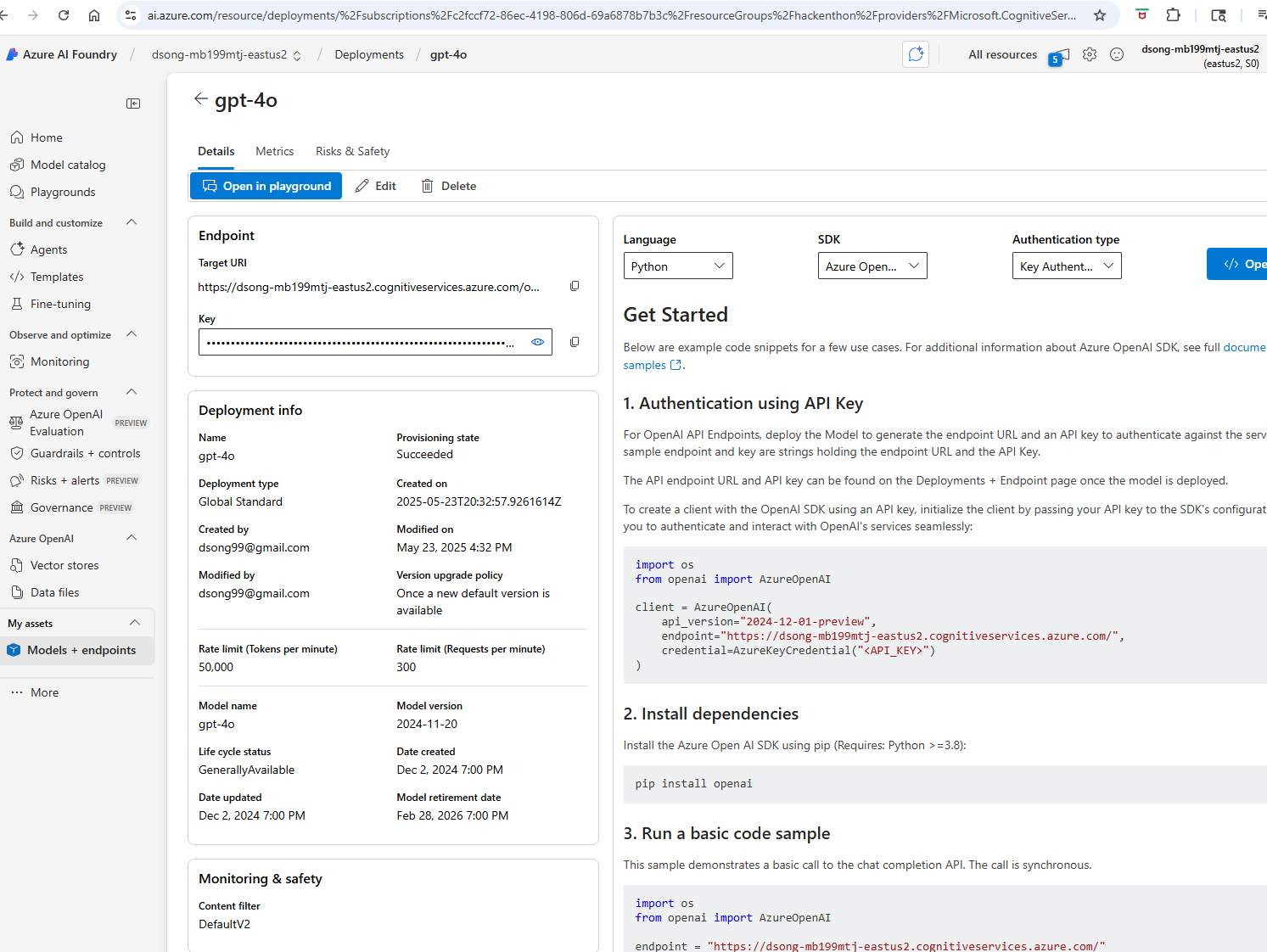


1. Fill in information, then click next, next, …, at the end click create button.



1. One the resource is created, it will be deployed. The deployment process will take a minute or two. Once it’s done, click on resources, this will bring to the resource page. Click on the resource you just created.

You can also get to the page by clicking **Model + endpoints** at the left end corner of the resource page.



1. On top of the page, you can see Endpoint and key, these two are needed in your python script, just copy those two values. On the right side of the page, it shows examples how to connect to Azure OpenAI.

# Use Python to create a chat bot

## Install Python packages

This example use a few Python packages, you will need to install them either use conda or pip: dotenv, langchain, langchain\_openai

## Create a file named: “.env”

Copy and paste the endpoint and key from resource page above to this file:

AZURE\_OPENAI\_API\_KEY='AxYnHXDtx\*\*\*'  
AZURE\_OPENAI\_ENDPOINT='https://dsong-\*\*\*version=2025-01-01-preview'

## Source code

from langchain\_openai import AzureChatOpenAI  
from dotenv import load\_dotenv  
from langchain.prompts import ChatPromptTemplate  
  
  
load\_dotenv()  
  
llm = AzureChatOpenAI(  
 azure\_deployment="gpt-4o", *# or your deployment* api\_version="2025-01-01-preview", *# or your api version*)  
  
user\_input=input('You:')  
  
prompt = ChatPromptTemplate(  
 {  
 ('system', ('You are a helpful AI assistant. Please assist the user')),  
 ('human', user\_input)  
 }  
)  
  
response = llm.invoke(prompt.format(user\_input=user\_input))  
print(response.content)

1. Note: load\_dotenv() loads the .env file and set properties in that file to the environment.

# Connect to Azure OpenAI

This example will show how to connect to AzureOpenAI without using langchain.

Required package: install openai

## Source Code

import os  
from openai import AzureOpenAI  
  
endpoint = 'https://dsong-mb199mtj-\*\*\* 2025-01-01-preview'  
model\_name = "gpt-4o"  
deployment = "gpt-4o"  
  
subscription\_key = 'AxYnHXDtxFCi1uuKKk3r7rk\*\*\*'  
api\_version = "2025-01-01-preview"  
  
client = AzureOpenAI(  
 api\_version=api\_version,  
 azure\_endpoint=endpoint,  
 api\_key=subscription\_key,  
)  
  
response = client.chat.completions.create(  
 messages=[  
 {  
 "role": "system",  
 "content": "You are a helpful assistant.",  
 },  
 {  
 "role": "user",  
 "content": "I am going to Paris, what should I see?",  
 }  
 ],  
 max\_tokens=10,  
 temperature=0,  
 top\_p=1.0,  
 model=deployment  
)  
  
print(response.choices[0].message.content)

# Generate two dataframe differences report from OpenAI

## Scouce code

import os  
from openai import AzureOpenAI  
  
endpoint = 'https://dsong-mb199mtj-eastus2.cognitiveservices.azure.com/openai/deployments/gpt-4o/chat/completions?api-version=2025-01-01-preview'  
model\_name = "gpt-4o"  
deployment = "gpt-4o"  
  
subscription\_key = 'AxYnHXDtxFCi1uuKKk3r7rknqyj4EeJTOnpRwL4tvzehsTXA2M1SJQQJ99BEACHYHv6XJ3w3AAAAACOGqphh'  
api\_version = "2025-01-01-preview"  
  
client = AzureOpenAI(  
 api\_version=api\_version,  
 azure\_endpoint=endpoint,  
 api\_key=subscription\_key,  
)  
import pandas as pd  
  
df1 = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})  
df2 = pd.DataFrame({'A': [1, 2, 5], 'B': [4, 7, 6]})  
  
comparison = df1.compare(df2)  
prompt = f"""  
Compare the following two dataframes and generate a report highlighting the differences:  
  
DataFrame 1:  
{df1.to\_string()}  
  
DataFrame 2:  
{df2.to\_string()}  
  
Generate a concise report summarizing the changes, including added, removed, and modified rows or values.  
"""  
  
  
response = client.chat.completions.create(  
 messages=[  
 {  
 "role": "system",  
 "content": "You are a helpful assistant.",  
 },  
 {  
 "role": "user",  
 "content": prompt,  
 }  
 ],  
 max\_tokens=200,  
 temperature=0,  
 top\_p=1.0,  
 model=deployment  
)  
  
  
*# Extract the generated report*print(response.choices[0].message.content)

## result

### Report: Comparison of DataFrame 1 and DataFrame 2

#### Summary of Differences:

1. \*\*Modified Values\*\*:

- In row index `1`, column `B`:

- \*\*DataFrame 1\*\*: Value is `5`

- \*\*DataFrame 2\*\*: Value is `7`

2. \*\*Modified Rows\*\*:

- Row index `2`:

- \*\*DataFrame 1\*\*: `[A=3, B=6]`

- \*\*DataFrame 2\*\*: `[A=5, B=6]`

3. \*\*No Added or Removed Rows\*\*:

- Both DataFrames have the same number of rows (3). No rows were added or removed.

#### Detailed Changes:

| Index | Column | DataFrame 1 Value | DataFrame 2 Value | Change Type |

|-------|--------|-------------------|-------------------|---------------|

| 1 | B | 5

# References:

1. OpenAI document <https://platform.openai.com/docs/overview>
2. Python connect to OpenAI: <https://www.youtube.com/watch?v=czvVibB2lRA>
3. Tutorial video on Azure OpenAI and GPT Models <https://www.youtube.com/watch?v=jQyYeYWD97I>
4. Tutorial video Getting Started with Azure OpenAI | GPT 4o <https://www.youtube.com/watch?v=H_1Ge6wxaaE>
5. How to setup Azure OpenAI service and use it in Python https://www.youtube.com/watch?v=50ZwmkCvE88
6. Free AI: <https://huggingface.co/>
7. Python langchain package: https://python.langchain.com/docs/integrations/chat/azure\_chat\_openai/