**How to Call Azure OpenAI on a Local Laptop Using VS Code and Python**

This document provides step-by-step instructions to set up and call Azure OpenAI services on your local machine using Python in Visual Studio Code (VS Code).

**1. Prerequisites**

Before starting, ensure the following:

1. **Azure Account**
   * Create an Azure account if you don’t already have one.
   * Deploy the Azure OpenAI Service in the Azure Portal.
   * Follow the [Azure OpenAI Quickstart Guide](https://learn.microsoft.com/en-us/azure/ai-services/openai/quickstart).
2. **Python Installed**
   * Install Python (3.7 or later) from [python.org](https://www.python.org/).
   * Verify the installation by running: python --version
3. **Visual Studio Code Installed**
   * Download and install [Visual Studio Code](https://code.visualstudio.com/).
4. **Azure CLI (Optional)**
   * Install the Azure CLI for managing Azure services.
   * [Installation Guide](https://learn.microsoft.com/en-us/cli/azure/install-azure-cli).

**2. Install Necessary Libraries**

1. Open VS Code and create a new project folder.
2. Create a Python virtual environment:

python -m venv venv

source venv/bin/activate # On Windows: venv\Scripts\activate

1. Install the required Python libraries:

pip install azure-ai-openai

pip install python-dotenv

**3. Set Up Azure OpenAI Service**

1. Log in to the [Azure Portal](https://portal.azure.com) and navigate to your Azure OpenAI resource.
2. Obtain the following details:
   * **Endpoint URL**: Found in the resource overview.
   * **API Key**: Found under "Keys and Endpoints" in the Azure OpenAI resource.

**4. Configure Environment Variables**

1. Create a .env file in your project directory to securely store sensitive information:

AZURE\_OPENAI\_ENDPOINT=https://<your-resource-name>.openai.azure.com/

AZURE\_OPENAI\_KEY=<your-api-key>

1. Add .env to .gitignore to prevent committing sensitive data.

**5. Write the Python Script**

Create a Python script file (e.g., main.py) in your project directory and add the following code:

import os

from azure.ai.openai import OpenAIClient

from azure.core.credentials import AzureKeyCredential

from dotenv import load\_dotenv

# Load environment variables

load\_dotenv()

# Azure OpenAI Configuration

endpoint = os.getenv("AZURE\_OPENAI\_ENDPOINT")

api\_key = os.getenv("AZURE\_OPENAI\_KEY")

# Initialize OpenAI Client

client = OpenAIClient(endpoint=endpoint, credential=AzureKeyCredential(api\_key))

# Call Azure OpenAI API

response = client.get\_completions(

engine="text-davinci-003", # Replace with your model deployment name

prompt="Write a poem about the ocean",

max\_tokens=50

)

# Print the response

print(response.choices[0].text.strip())

**6. Run the Script**

1. Open the terminal in VS Code.
2. Run the Python script:

python main.py

**7. Debug and Test**

1. Use VS Code’s built-in debugger to inspect variables and trace execution.
2. Modify the prompt or parameters like max\_tokens to test different use cases.

**8. Optional Enhancements**

* **Logging**: Use Python's logging module for better traceability.
* **Error Handling**: Wrap API calls in try-except blocks to handle errors gracefully.
* **Encapsulation**: Encapsulate API calls in reusable functions for cleaner code.

**Resources**

* [Azure OpenAI Documentation](https://learn.microsoft.com/en-us/azure/ai-services/openai/)
* [Azure SDK for Python](https://learn.microsoft.com/en-us/python/api/overview/azure/ai/openai?view=azure-python)