

Practical - 3

Azure and Generative AI

Introduction

Microsoft Azure is a leading cloud computing platform that provides a vast ecosystem of services, tools, and infrastructure, empowering organizations to develop, deploy, and manage applications and services through a global network of Microsoft-managed data centers. Azure provides comprehensive solutions across various computing models, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), catering to a broad spectrum of business and technological requirements, from basic storage solutions to advanced artificial intelligence and machine learning deployments.

Fundamental Azure Concepts

- **Resource Groups:** Azure resource groups are logical containers used to manage and organize related Azure resources. A resource group serves as a unit of management, allowing users to deploy, update, and delete all resources within the group in a coordinated manner. This facilitates efficient management, simplifies access control, and streamlines cost tracking. All resources within a resource group share the same lifecycle, ensuring consistent management and deployment.
- **Azure Services:** The Azure platform offers a wide range of services designed to meet diverse computing needs. Key services relevant to AI and machine learning include:
 - **Azure OpenAI Service:** This service enables organizations to leverage state-of-the-art OpenAI models, such as GPT-3.5, GPT-4, and various embedding models, directly within the Azure environment. It provides enterprise-grade security, compliance certifications, and global availability, making it suitable for a range of commercial applications. The Azure OpenAI Service ensures data privacy and security, with Microsoft guaranteeing that customer data is not used to train OpenAI models.
 - **Azure AI Studio/Foundry:** Azure AI Studio (formerly AI Foundry) is a unified platform for building, evaluating, and deploying AI models. It supports both proprietary models available through Azure and open-source models, providing a flexible environment for AI innovation. Azure AI Studio streamlines the AI development lifecycle, offering tools for data preparation, model training, evaluation, and deployment. This facilitates the integration of AI capabilities into various applications and services.

- **Azure Machine Learning:** A cloud-based platform for developing, training, and deploying machine learning models. It supports various machine learning frameworks and languages, offering automated machine learning (AutoML) capabilities to simplify model creation.

Deployment Process and Troubleshooting

1. **Resource Group Creation:** The first step in utilizing Azure services involves creating a resource group via the Azure portal. The resource group serves as a logical container for all related resources, such as the Azure OpenAI instance, virtual networks, and storage accounts. When creating a resource group, it is necessary to specify a region, which determines the physical location where resources are deployed.
2. **Service Provisioning:** Once the resource group is created, the required services, such as Azure OpenAI, can be provisioned within it. Provisioning Azure OpenAI involves selecting a pricing tier (e.g., S0), specifying the region for deployment, and configuring other settings as needed. The pricing tier determines the performance characteristics and cost of the service.
3. **Model Deployment:** After provisioning the Azure OpenAI service, the next step is to deploy a specific model, such as GPT-3.5-turbo. Model deployment involves selecting the desired model and creating a deployment with a unique name. Azure OpenAI Service allows multiple deployments within a single instance, enabling users to experiment with different configurations and versions of models.
4. **Subscription and Eligibility Issues:** As seen in the initial error (UserError: Error occurred when subscribing to Marketplace: Marketplace Subscription purchase eligibility check failed), deploying certain models, especially those from the Azure Marketplace (e.g., Llama-3.3-70B-Instruct), requires a paid Azure subscription. This error indicates that the user's current subscription (likely a free trial) is not eligible to purchase and deploy the selected model. Paid subscriptions provide the necessary entitlements and resources for deploying such models, which often have specialized licensing and infrastructure requirements.
 - **Troubleshooting Subscription Issues:** Resolving subscription-related errors involves upgrading to a Pay-As-You-Go subscription or another eligible paid plan. This ensures that the account has the required permissions and billing setup to deploy and use marketplace models.
5. **Connecting to Models:** Once a model is successfully deployed, it can be accessed via an API endpoint. Azure provides an API endpoint and access keys (or Azure Active Directory

authentication) to secure access to the deployed model. These credentials are used to authenticate API requests and authorize access to the model.

Connecting via VS Code: A Step-by-Step Guide

Visual Studio Code (VS Code) is a versatile and widely used code editor, favored for its extensibility, debugging capabilities, and integration with various development tools. Integrating VS Code with Azure OpenAI allows developers to efficiently develop, test, and deploy AI applications.

1. **Environment Setup:** Creating a virtual environment using tools like conda or venv is crucial for managing project-specific dependencies. This prevents conflicts between different Python projects and ensures reproducibility.

- o **Creating a Virtual Environment (using conda):**

```
bash
```

```
conda create --name myenv python=3.9
```

```
conda activate myenv
```

2. **Installing Dependencies:** Install the necessary Python packages using pip. The openai package facilitates interaction with the Azure OpenAI API, while python-dotenv manages environment variables.

```
bash
```

```
pip install openai python-dotenv
```

3. **Environment Variables:** Store API keys and endpoints as environment variables to avoid hardcoding them directly in the code. This enhances security and makes the code more portable. Create a .env file in the project directory and define the required variables.

```
text
```

```
AZURE_OPENAI_ENDPOINT="YOUR_AZURE_OPENAI_ENDPOINT"
```

```
AZURE_OPENAI_API_KEY="YOUR_AZURE_OPENAI_API_KEY"
```

4. **Coding the Interaction:** Write Python code to interact with the Azure OpenAI service using the openai library. The code should load the environment variables, authenticate with Azure OpenAI, and send requests to the deployed model.

```
python
```

```
import os
```

```

import openai

from dotenv import load_dotenv

# Load environment variables from .env file

load_dotenv()

openai.api_type = "azure"

openai.api_base = os.getenv("AZURE_OPENAI_ENDPOINT")

openai.api_version = "2023-05-15"

openai.api_key = os.getenv("AZURE_OPENAI_API_KEY")

def get_openai_response(prompt, deployment_name="your-deployment-name"):
    response = openai.ChatCompletion.create(
        engine=deployment_name,
        messages=[
            {"role": "system", "content": "You are a helpful AI assistant."},
            {"role": "user", "content": prompt}
        ]
    )
    return response['choices'][0]['message']['content']

if __name__ == "__main__":
    user_prompt = input("Enter your prompt: ")
    ai_response = get_openai_response(user_prompt)
    print("AI Response:", ai_response)

```

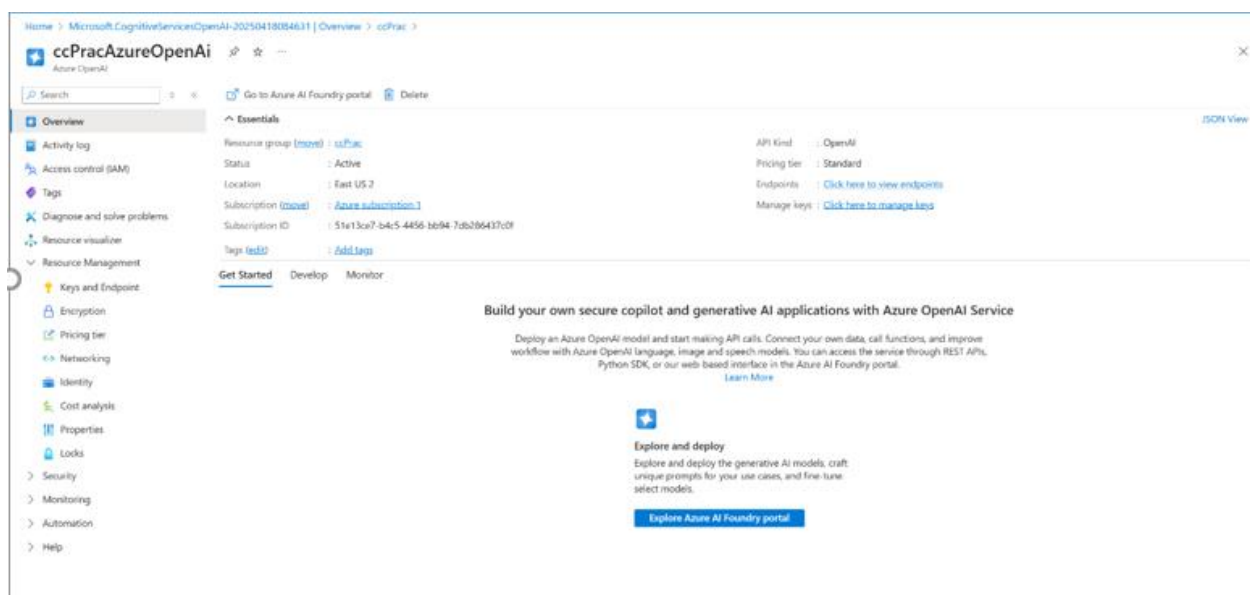
Detailed Dependency Explanation

- **openai:** This Python library provides an interface for interacting with the OpenAI API. It simplifies tasks such as authentication, request formatting, and response handling. The openai package abstracts away the complexities of making raw HTTP requests, allowing developers to focus on application logic.
- **python-dotenv:** This package enables the loading of environment variables from a .env file. It enhances security by keeping sensitive information (API keys, database passwords) separate from the code. The python-dotenv package parses the .env file and makes the variables available through os.environ.

Conclusion

Azure offers a comprehensive and scalable platform for deploying and utilizing generative AI models. While challenges such as subscription limitations and deployment issues can arise, a thorough understanding of the Azure ecosystem, coupled with effective resource management and dependency handling, enables developers to harness the full potential of AI in their applications. Integrating these services with VS Code streamlines the development workflow, facilitating efficient integration, testing, and deployment of AI-powered functionalities. Proper management of dependencies, secure handling of API keys, and a structured deployment process are essential for successful AI application development on Azure.

Azure Dashboard -



Welcome to Azure OpenAI Service

Explore the generative AI models and craft unique prompts for your use cases.

Resource configuration

Name	Subscription	Subscription ID
ccPracAzureOpenAI	Azure subscription 1	51e13ce7-b4c5-4456-bb94-7db296437cdf
API key 1	Resource group	Pricing tier
	ccPrac	Standard 50
API key 2	Azure OpenAI Service endpoint	Location
	https://ccpracazureopenai.openai.azure.com/	eastus2

Get started

Assistants playground

Speed up development of GPT-powered AI Assistants with prebuilt conversation state management and customization tools.

[Try it now](#)

Chat playground

Design a customized AI assistant using ChatGPT. Experiment with GPT-3.5-Turbo and GPT-4 models.

[Try it now](#)

Bring your own data

Ground your own data on advanced AI models to create conversational copilots that aid user comprehension, task completion, and decision-making.

[Try it now](#)

Completions playground

Experiment with completions models for use cases such as summarization, content generation, and classification.

[Try it now](#)

Azure OpenAI Service help

What is Azure OpenAI Service?

Azure OpenAI Service provides access to OpenAI's models with the security and enterprise capabilities of Azure.

[Learn more](#)

Relevant resources

Chat playground

[View code](#) [Deploy](#) [Import](#) [Export](#)

Setup

Deployment

gpt-4 (version:turbo-2024-04-09)

Please select the 'Apply changes' button below.

Give the model instructions and context

Apply changes

Generate prompt

+ Add section

> Add your data

> Parameters

Chat history

The "Seven Wonders of the World" can refer to different sets of remarkable structures from various points in history. Here are two of the most popular lists:

- The Seven Wonders of the Ancient World** (which includes wonders from the Mediterranean and Middle Eastern region, largely from the Hellenistic period):
 - The Great Pyramid of Giza in Egypt (the only ancient wonder still standing today)
 - The Hanging Gardens of Babylon in Iraq
 - The Statue of Zeus at Olympia in Greece
 - The Temple of Artemis at Ephesus in Turkey
 - The Mausoleum at Halicarnassus in Turkey
 - The Colossus of Rhodes in Greece
 - The Lighthouse of Alexandria in Egypt
- The New7Wonders of the World** (a modern list compiled by a global poll organized by the New7Wonders Foundation in 2007):
 - The Great Wall of China in China
 - Petra in Jordan
 - Christ the Redeemer in Brazil
 - Machu Picchu in Peru
 - Chichen Itza in Mexico
 - The Roman Colosseum in Italy
 - The Taj Mahal in India

These lists celebrate both ancient and more modern masterpieces of construction and cultural heritage.

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

308/128000 tokens to be sent

