Labs Data & AI Innovation Day

Lab 1: Prepare environment

This document explains how to create databases and containers that support vector search in a Cosmos DB for NoSQL account

# Deploy Azure OpenAI models

Follow these steps to deploy the Azure OpenAI models (GPT 3.5 Turbo and Text-embeddingt-ada-002)

* Login to the Azure Portal
* Connect to the Azure OpenAI account
* In the left menu, select “Model deployments”
* Click “Manage deployments”
* Select the “Deployments” section
* Click on the “Create deployment” button
* Select the “text-embedding-ada-002” model in the dropdown list
* In the deployment name, type text-embedding-ada-002
* Click on the “Create” button to deploy the model

A screenshot of a computer

Description automatically generated

* Repeat the previous steps to deploy the gpt-3.5-turbo model

## Get Azure OpenAI key

* Connect to Azure Portal
* Connect to Azure OpenAI account
* In the left navigation menu, select “Keys and Endpoint”
* Key 1 and endpoint are the values you will need to add to your .env file

A screenshot of a computer

Description automatically generated

# Prepare Python environment

## Create environment variables

1. Create a “labs” folder on your local machine
2. Create a “.env” file at the root of the labs folder
3. Add the following environment variables

|  |  |  |
| --- | --- | --- |
| AZURE\_COSMOSDB\_NOSQL\_ENDPOINT | The name of your Cosmos DB for NoSQL | cosmos-nosql-001documents.azure.com |
| AZURE\_COSMOSDB\_NOSQL\_KEY | Cosmos DB key | xxxx |
| AZURE\_COSMOSDB\_NOSQL\_DATABASE\_NAME | Database name | Database\_teamXX |
| AZURE\_COSMOSDB\_NOSQL\_DATABASE\_NAME | Database name | database\_<team\_name>, e.g. database\_team01 |
| AZURE\_COSMOSDB\_NOSQL\_VECTORS\_CONTAINER\_NAME | Vectors container name | Vectors |
| AZURE\_OPENAI\_ENDPOINT | Azure OpenAI account url | https://<team\_name>openai.openai.azure.com/ |
| AZURE\_OPENAI\_API\_KEY | Azure OpenAI account key |  |
| AZURE\_OPENAI\_EMBEDDING\_MODEL | Name of your embedding model deployment | Defaults to text-embedding-ada-002 |
| AZURE\_OPENAI\_CHAT\_MODEL | Name of your chat model deployment | Defaults to gpt-35-turbo |
| AZURE\_OPENAI\_API\_VERSION | API version | 2024-02-01 |

A screenshot of a computer

Description automatically generated

## Create virtual environment

1. Create a “requirements.txt” file at the root of the “labs” folder
2. Add the following libraries

python-dotenv

tenacity

ipykernel

matplotlib

plotly

scikit-learn

openai

azure-cosmos

streamlit

pymongo

tiktoken

azure-identity

1. Open a command prompt and navigate to the “labs” folder
2. Create a virtual environment with this command: python -m venv .venv
3. Activate the virtual environment with .venv\scripts\activate
4. Install the required libraries with pip install -r requirements.txt

A computer screen shot of a computer code

Description automatically generated

# Create a container that supports vector search

In this section, we will use a Python notebook to create a container that supports vector search

## Create product container

1. Create a “Lab1” folder in the “Labs” foler
2. Open Visual Studio Code
3. In the “Lab1” folder, create a new Jupyter notebook called “create\_container.ipynb”
4. Create a new cell and add the following content

from azure.cosmos import CosmosClient, PartitionKey

from azure.cosmos.cosmos\_client import ThroughputProperties

from dotenv import load\_dotenv

load\_dotenv("..\.env")

COSMOS\_DB\_ENDPOINT = os.getenv('AZURE\_COSMOSDB\_NOSQL\_ENDPOINT')

COSMOS\_DB\_KEY = os.getenv('AZURE\_COSMOSDB\_NOSQL\_KEY')

DATABASE\_NAME = os.getenv('AZURE\_COSMOSDB\_NOSQL\_DATABASE\_NAME')

CONTAINER\_NAME = os.getenv('AZURE\_COSMOSDB\_NOSQL\_CONTAINER\_NAME')

OFFER\_THROUGHPUT = 1000

throughput\_properties = ThroughputProperties(auto\_scale\_max\_throughput=OFFER\_THROUGHPUT)

indexing\_policy = {

    "includedPaths": [

        {"path": "/\*"},

    ],

    "excludedPaths": [

        {"path": "/\"\_etag\"/?"},

        {"path": "/embedding/\*"}

    ],

    "vectorIndexes": [

        {

            "path": "/embedding",

            "type": "quantizedFlat"

        }

    ]

}

embedding\_policy = {

    "vectorEmbeddings": [

        {

            "path": "/embedding",

            "dataType": "float32",

            "distanceFunction": "cosine",

            "dimensions": 384

        }

    ]

}

print("Getting database..")

client = CosmosClient(COSMOS\_DB\_ENDPOINT, COSMOS\_DB\_KEY)

database = client.get\_database\_client(DATABASE\_NAME)

*# Create "products" container*

print(f"Creating '{CONTAINER\_NAME}' container..")

container = database.create\_container\_if\_not\_exists(

    id=CONTAINER\_NAME,

    partition\_key=PartitionKey(path="/id"),

    indexing\_policy=indexing\_policy,

    vector\_embedding\_policy=embedding\_policy,

    offer\_throughput=throughput\_properties

)

print(f"'{CONTAINER\_NAME}' container created.")

1. In the top-right corner, click on “Select kernel”
2. Select “Python environment” and select the recommended environment (that should point to the .venv environment that you created in the preview section)  
     
   A screenshot of a computer program

   Description automatically generated
3. In the menu bar, click on “Run all”  
     
   A screen shot of a computer

   Description automatically generated  
     
   A screenshot of a computer program

   Description automatically generated
4. Verify in the Azure portal that the “products” container has been successfully created.

## A screenshot of a computer Description automatically generated