

Київський національний університет ім. Тараса Шевченка

Кафедра мережевих та інтернет технологій

## Лабораторна робота №4

**Дисципліна:** Хмарні технології

**Тема:** Створення класифікатора документів із застосуванням Azure AI

Підготувала студентка групи МІТ-31

Пась Олександра

### Хід виконання роботи

Для початку було створено Azure AI Language сервіс та отримано ключ доступу та ендпоінт, щоб мати доступ до сервісу із застосунку.

**Create Language** ...

what customers think of your brand. Find topic-relevant phrases using key phrase extraction and identify the language of the text with language detection. Detect and categorize entities in your text with named entity recognition.

[Learn more](#)

**Project Details**

Subscription \* ⓘ

Resource group \* ⓘ  [Create new](#)

**Instance Details**

Region ⓘ


Name \* ⓘ  ✓

Pricing tier \* ⓘ

[View full pricing details](#)

рис. 4.1 - Налаштування Azure AI Language сервісу

✓ Your deployment is complete

 Deployment name : TextAnalyticsCreate-20250306195838

Subscription : Azure subscription 1

Resource group : lab1

Start time : 3/6/2025, 8:03:38 PM

Correlation ID : 21188cff-8c51-40f1-aaca-fb2c26c3468d

> Deployment details

✓ Next steps

[Go to resource](#)

рис. 4.2 - успішне створення

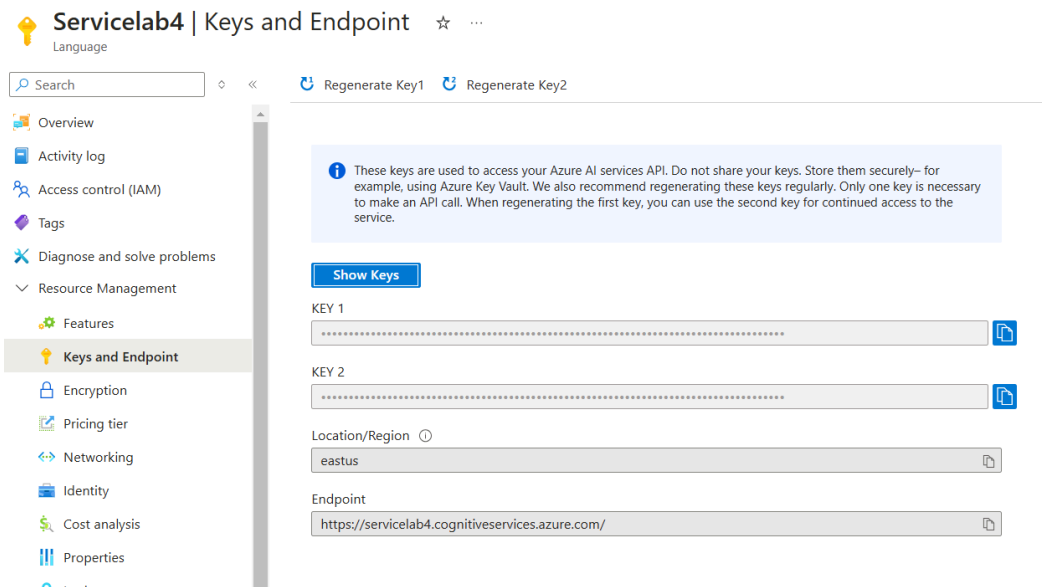


рис. 4.3 - Отримання ключа доступу та ендпоінту сервісу

Для створення контейнерів для вхідних та результуючих даних, необхідно створити Azure Storage Account.

#### Create a storage account

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *	Azure subscription 1
Resource group *	lab1
<a href="#">Create new</a>	
<b>Instance details</b>	
Storage account name *	storageaccountpom
Region *	(US) East US
<a href="#">Deploy to an Azure Extended Zone</a>	
Primary service	Azure Blob Storage or Azure Data Lake Storage Gen 2
Performance *	<input checked="" type="radio"/> Standard: Recommended for most scenarios (general-purpose v2 account)
	<input type="radio"/> Premium: Recommended for scenarios that require low latency.
Redundancy *	Geo-redundant storage (GRS)
	<input checked="" type="checkbox"/> Make read access to data available in the event of regional unavailability.

рис. 4.4 - створення Storage account

Your deployment is complete

Deployment name: storageaccountpom\_1741284915144  
Subscription: [Azure subscription 1](#)  
Resource group: [lab1](#)

Start time: 3/6/2025, 8:15:26 PM  
Correlation ID: 8c1dc9ab-fd92-42a3-95f8-0f534870af28

Deployment details

Next steps

[Go to resource](#)

рис. 4.5 - успішне створення

Для створення контейнерів потрібно обрати секцію Containers та створити два контейнери source та destination. Після цього отримати ключ доступу.

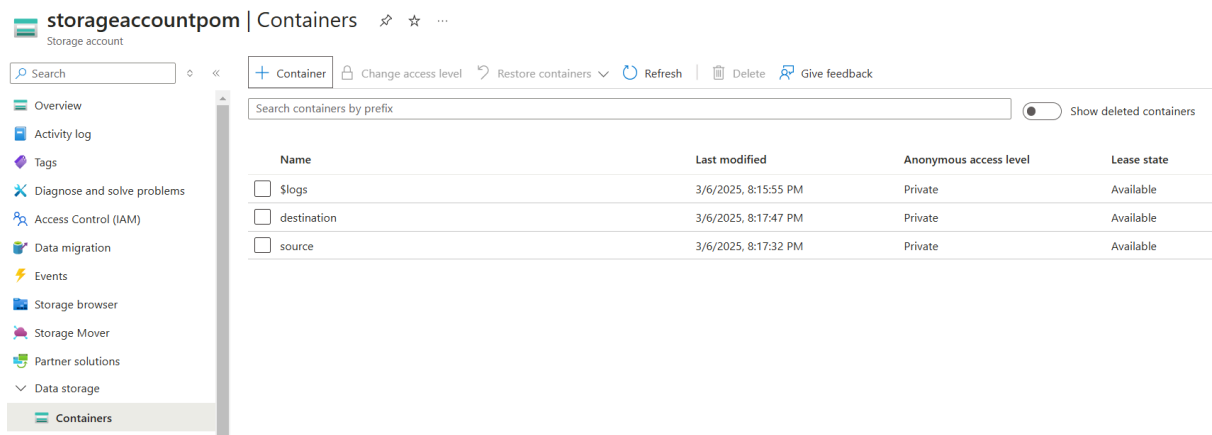


рис. 4.6 - створені контейнери

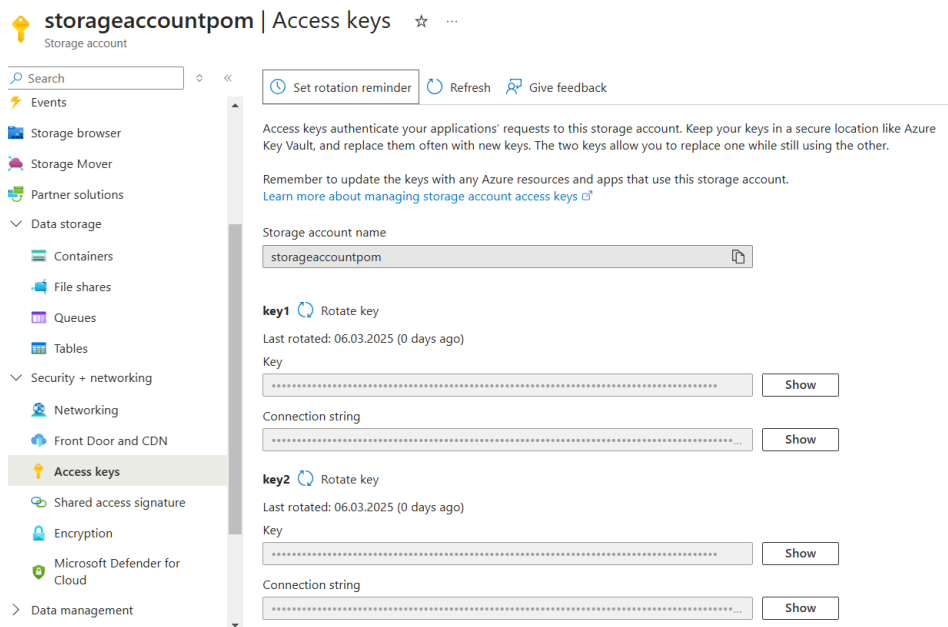


рис. 4.7 - ключі доступу

Наступним кроком було створення проєкту із застосуванням шаблону Azure Functions.

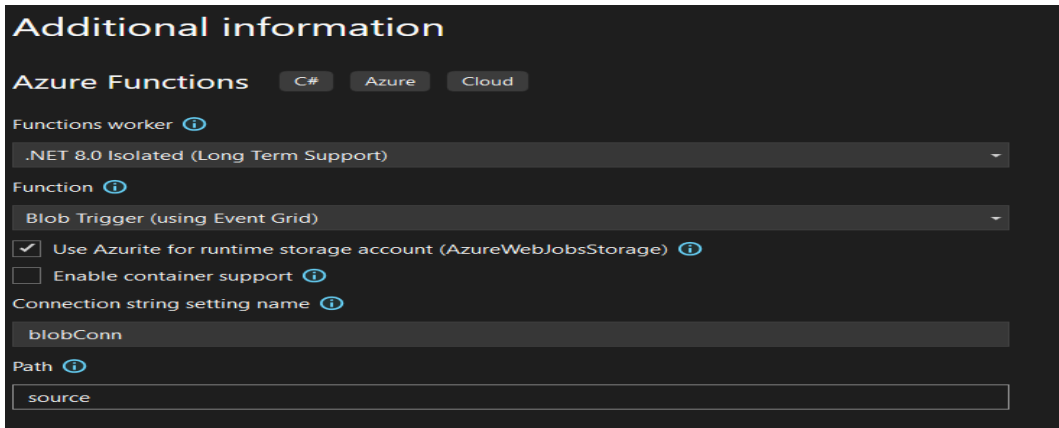


рис.4.8 - створення проєкту

На наступному кроці додали необхідні ключі та значення у файл local.settings.json.

```
{
  "IsEncrypted": false,
  "Values": {
    "AzureWebJobsStorage": "UseDevelopmentStorage=true",
    "FUNCTIONS_WORKER_RUNTIME": "dotnet-isolated",
    "sourceContainerName": "source",
    "targetContainerName": "destination",
    "blobConn": "DefaultEndpointsProtocol=https;AccountName=storageaccountpom;AccountKey=nBzETJsvMPSzm9mhUugKCF5h",
    "textAnalyticsEndpoint": "https://servicelab4.cognitiveservices.azure.com/",
    "textAnalyticsLocation": "eastus",
    "textAnalyticsKey": "8sJtSjhCQSt3l86tTpdHDhB3eFdLbAwarDbZbsVPuAlUkCKjZq8HJQQJ99BCACyEbjFXJ3w3AAAaACOGgWte"
  }
}
```

рис. 4.9 - зміни у local.settings.json

Також було додано підтримку клієнта Azure та сервісу зберігання BLOB-об'єктів у наш додаток, оновивши файл Program.cs.

```
using Azure;
using Microsoft.Azure.Functions.Worker;
using Microsoft.Azure.Functions.Worker.Builder;
using Microsoft.Extensions.Azure;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;

var builder = FunctionsApplication.CreateBuilder(args);

builder.ConfigureFunctionsWebApplication();
builder.Services.
    AddAzureClients(b =>
    {
        b.AddBlobServiceClient(Environment.GetEnvironmentVariable("blobConn"));
        var endpoint = new Uri(Environment.
            GetEnvironmentVariable("textAnalyticsEndpoint"));
        var credential = new AzureKeyCredential(Environment.
            GetEnvironmentVariable("textAnalyticsKey"));
        b.AddTextAnalyticsClient(endpoint, credential);
    });

builder.Build().Run();
```

рис. 4.10 - зміни в program.cs

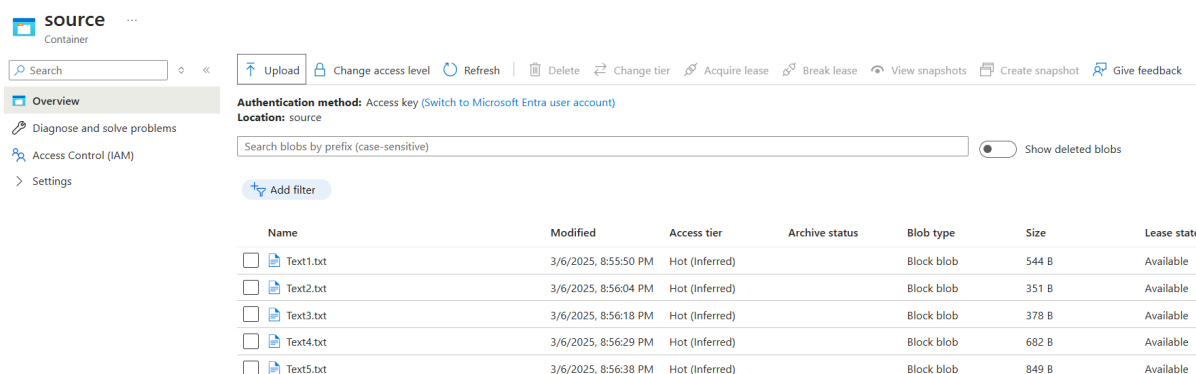
```

private readonly TextAnalyticsClient _textAnalyticsClient;
private readonly BlobServiceClient _blobServiceClient;
private readonly BlobContainerClient _blobSourceContainerClient;
private readonly BlobContainerClient _blobDestinationContainerClient;
0 references
public LangBasedDocumentClassifier(ILogger<LangBasedDocumentClassifier> logger,
TextAnalyticsClient textAnalyticsClient,
BlobServiceClient blobServiceClient)
{
    _logger = logger;
    _textAnalyticsClient = textAnalyticsClient;
    _blobServiceClient = blobServiceClient;
    _blobSourceContainerClient = _blobServiceClient.GetBlobContainerClient(Environment.
GetEnvironmentVariable("sourceContainerName"));
    _blobDestinationContainerClient = _blobServiceClient.GetBlobContainerClient(Environment.
GetEnvironmentVariable("targetContainerName"));
}
[Function(nameof(LangBasedDocumentClassifier))] // BlobTriggerSource.LogsAndContainerScan
0 references
public async Task Run([BlobTrigger("source/{name}", Source =
BlobTriggerSource.LogsAndContainerScan, Connection = "blobConn")] Stream stream, string name)
{
    using var blobStreamReader = new StreamReader(stream);
    var content = await blobStreamReader.ReadToEndAsync();
    _logger.LogInformation($"C# Blob Trigger (using Event Grid) processed blob\n Name: {name}" +
"\n Data: { content} ");
    var detectedLanguage = await _textAnalyticsClient.DetectLanguageAsync(content);
    var languageName = detectedLanguage.Value.Name;
    _logger.LogInformation($"Detected language: {languageName}");
    string targetBlobName = $"{languageName}/{name}";
    BlobClient blobClient = _blobDestinationContainerClient.GetBlobClient(targetBlobName);
    byte[] byteArray = Encoding.UTF8.GetBytes(content);
    await blobClient.UploadAsync(new MemoryStream(byteArray));
    _logger.LogInformation($"Upload blob - {targetBlobName} in the " +
$"{Environment.GetEnvironmentVariable("targetContainerName")} container.");
    await _blobSourceContainerClient.DeleteBlobIfExistsAsync(name);
    _logger.LogInformation($"Deleted blob - {name} from " +
$"{Environment.GetEnvironmentVariable("sourceContainerName")} container.");
}

```

рис. 4.11 - зміни в Function1.cs

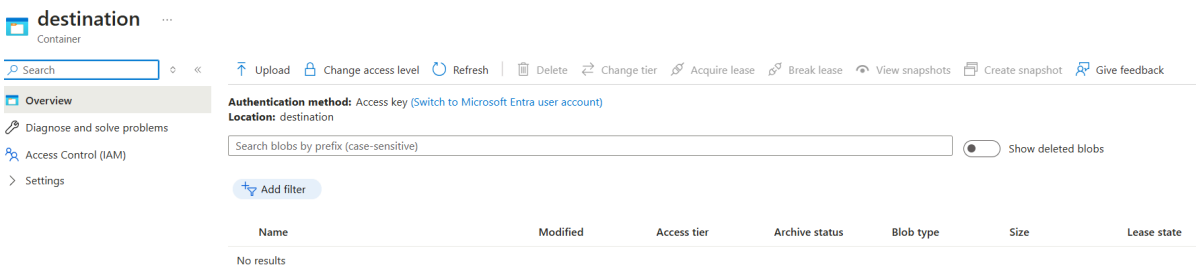
Після цього в контейнер source було додано файли з різними мовами (варіант 1: Ukrainian, English, French, Greek, Hindi).



Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
<input type="checkbox"/> Text1.txt	3/6/2025, 8:55:50 PM	Hot (Inferred)		Block blob	544 B	Available
<input type="checkbox"/> Text2.txt	3/6/2025, 8:56:04 PM	Hot (Inferred)		Block blob	351 B	Available
<input type="checkbox"/> Text3.txt	3/6/2025, 8:56:18 PM	Hot (Inferred)		Block blob	378 B	Available
<input type="checkbox"/> Text4.txt	3/6/2025, 8:56:29 PM	Hot (Inferred)		Block blob	682 B	Available
<input type="checkbox"/> Text5.txt	3/6/2025, 8:56:38 PM	Hot (Inferred)		Block blob	849 B	Available

рис. 4.12 - вміст контейнера source

Контейнер destination перед початком роботи функції порожній



Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
No results						

рис. 4.13 - вміст контейнера destination

```
Azure Functions Core Tools
Core Tools Version:      4.0.6821 Commit hash: N/A +c09a2033faa7ecf51b3773308283af0ca9a99f83 (64-bit)
Function Runtime Version: 4.1036.1.23224

[2025-03-06T18:58:23.888Z] Found C:\Users\Dell\source\repos\Lab4\Lab4\Lab4.csproj. Using for user secrets file configura
tion.
[2025-03-06T18:58:45.444Z] Azure Functions .NET Worker (PID: 12432) initialized in debug mode. Waiting for debugger to a
ttach...
[2025-03-06T18:58:45.659Z] Worker process started and initialized.

Functions:

    LangBasedDocumentClassifier: blobTrigger

For detailed output, run func with --verbose flag.
[2025-03-06T18:58:49.271Z] Executing 'Functions.LangBasedDocumentClassifier' (Reason='New blob detected(LogsAndContainer
Scan): source/Text1.txt', Id=62751f57-7e45-4586-93e2-bf283d2ca677)
[2025-03-06T18:58:49.280Z] Trigger Details: MessageId: 6e9b4367-2bbb-42a6-a2d3-49a0ef4faa09, DequeueCount: 1, InsertedOn
: 2025-03-06T18:58:48.000+00:00, BlobCreated: 2025-03-06T18:55:50.000+00:00, BlobLastModified: 2025-03-06T18:55:50.000+0
0:00
[2025-03-06T18:58:49.361Z] Executing 'Functions.LangBasedDocumentClassifier' (Reason='New blob detected(LogsAndContainer
Scan): source/Text2.txt', Id=5770d6c8-d4c1-43e1-a4fe-b22a290a300d)
[2025-03-06T18:58:49.364Z] Trigger Details: MessageId: 6dc8f9e6-8850-41a0-953c-c8611017d526, DequeueCount: 1, InsertedOn
: 2025-03-06T18:58:48.000+00:00, BlobCreated: 2025-03-06T18:56:04.000+00:00, BlobLastModified: 2025-03-06T18:56:04.000+0
```

рис. 4.14 - запуск проекта

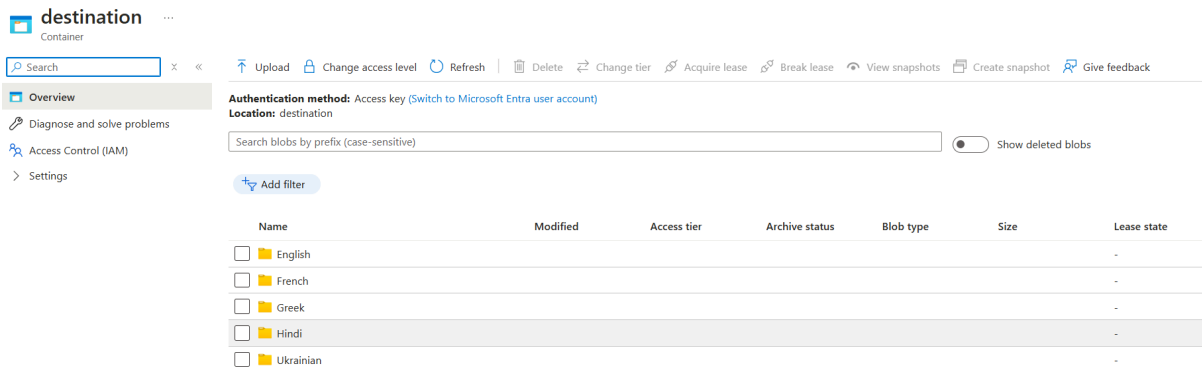


рис. 4.15 - вміст контейнера destination після запуску програми

**Висновок:** Під час виконання цієї лабораторної роботи було створено Azure Storage Account, два контейнери source та destination, класифікатор документів із використанням Azure Function. Також я створила п'ять текстових файлів різними мовами (Ukrainian, English, French, Greek, Hindi) та завантажила їх у контейнер source. Запустивши на виконання класифікатор документів було перевірено точність класифікації для кожної мови.