Microsoft 365 Mailbox Attachment Processor

A .NET8 C# Application to Process Microsoft 365 Email Messages and Attachments

Introduction

Recently, I was asked how attachments from a Microsoft 365 mailbox could be automatically pushed into an Azure storage file share, so that the attachments can be made accessible to an onward process which needs to be executed on an Azure Virtual Machine. Whilst there are many ways this can be achieved; I decided to create a C# Console application to process the messages and attachments from Microsoft 365 mailbox inbox folder.

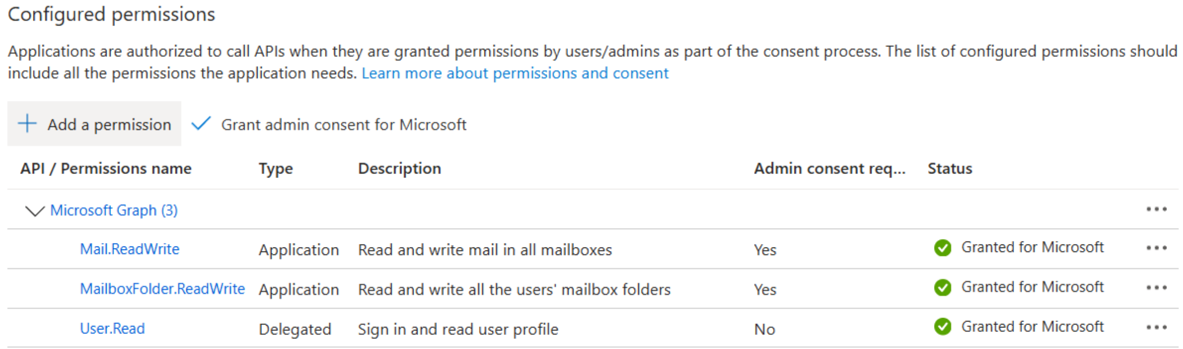
Source Code

The source code for this solution can be found in my GitHub repo here.

Dependencies

There are several dependencies for this to work, these are described in the list below.

- A Microsoft Entra ID registered application, with the following delegated application permissions:



Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (3)				
Mail.ReadWrite	Application	Read and write mail in all mailboxes	Yes	✓ Granted for Microsoft
MailboxFolder.ReadWrite	Application	Read and write all the users' mailbox folders	Yes	✓ Granted for Microsoft
User.Read	Delegated	Sign in and read user profile	No	✓ Granted for Microsoft

- An application secret (this can also be a certificate if needed)
- Install the Microsoft ExchangeOnlineManagement PowerShell tools
- Create a Microsoft Exchange Online application policy to allow the application access to the mailbox

```
# Connect to Exchange Online
Connect-ExchangeOnline -UserPrincipalName [Your Exchange Online Admin UPN] [-ShowBanner:$false]

# Create the app policy
New-ApplicationAccessPolicy -AppId [Your application ID] -PolicyScopeGroupId [Full email address of the mailbox] -AccessRight RestrictAccess -Description "Restrict the Mailbox Processor app.."
```

Reference: Limiting application permissions to specific Exchange Online mailboxes - Microsoft Graph | Microsoft Learn

- Create an Azure storage account
- Create an Azure storage account file share

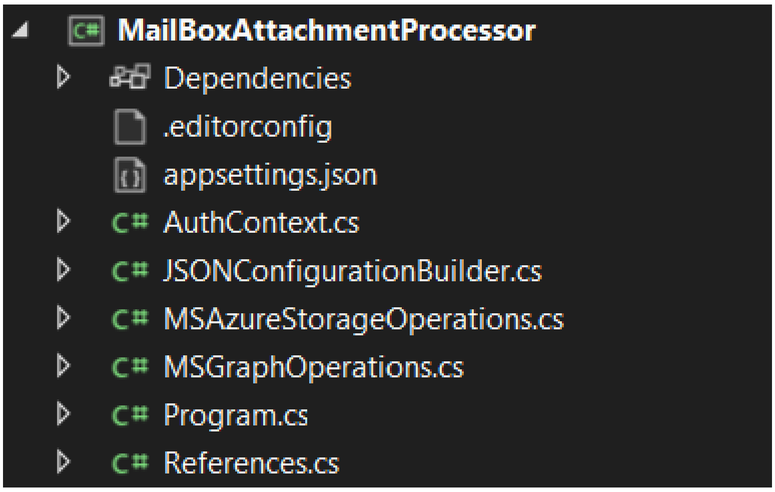
Nuget Packages

The following Nuget packages are a dependency as defined in the project settings.

```
<ItemGroup>
  <PackageReference Include="Azure.Core" Version="1.44.1" />
  <PackageReference Include="Azure.Identity" Version="1.13.1" />
  <PackageReference Include="Azure.Storage.Files.Shares" Version="12.21.0" />
  <PackageReference Include="Microsoft.Extensions.Configuration.Binder" Version="9.0.0" />
  <PackageReference Include="Microsoft.Extensions.Configuration.Json" Version="9.0.0" />
  <PackageReference Include="Microsoft.Graph" Version="5.63.0" />
  <PackageReference Include="Microsoft.Graph.Core" Version="3.2.1" />
  <PackageReference Include="Microsoft.Identity.Client" Version="4.66.2" />
</ItemGroup>
```

Mailbox Processor Application

The mailbox processor application consists of the following C# Classes and an appsettings.json file.



MailBoxAttachmentProcessor

- Dependencies
- .editorconfig
- appsettings.json
- AuthContext.cs
- JSONConfigurationBuilder.cs
- MSAzureStorageOperations.cs
- MSGraphOperations.cs
- Program.cs
- References.cs

File Name	Purpose
AuthContext.cs	A C# Class representing the authentication context for the application
JSONConfigurationBuilder.cs	A C# Class building the configuration from appsettings.json into the application context
MSAzureStorageOperations.cs	A C# Class with a method to stream the attachment to Azure Storage File Share
MSGraphOperations.cs	A C# Class with methods to work with the Microsoft Graph API e.g. read/move messages and attachments and folders
Program.cs	A C# program, the core of the application
Reference.cs	A C# Class to store the appsettings that are referenced by the application
Appsettings.json	The configuration settings for the application

Application Settings

The application settings have been described below.

```
{
  "AppSettings": {
    "MailFolderName": "[The mailbox folder to target to read the messages]",
    "MailEmailAddress": "[The mailbox email address]",
    "MailSubjectSearchString": "[The subject search string for each mail 
message]",
    "ProcessedMessagesFolderName": "[Process message mailbox folder name]",
    "AzureStorageConnectionString": "[The Azure storage connection string] ",
    "AzureStorageFileShareName": "[Azure storage file share name]",
    "MSEntraApplicationClientId": "[Microsoft Entra ID Application Id]",
    "MSEntraApplicationSecret": "[Microsoft Entra ID Application Secret]",
    "MSEntraApplicationTenantId": "[Microsoft Entra ID Tenant Id]"
  }
}
```

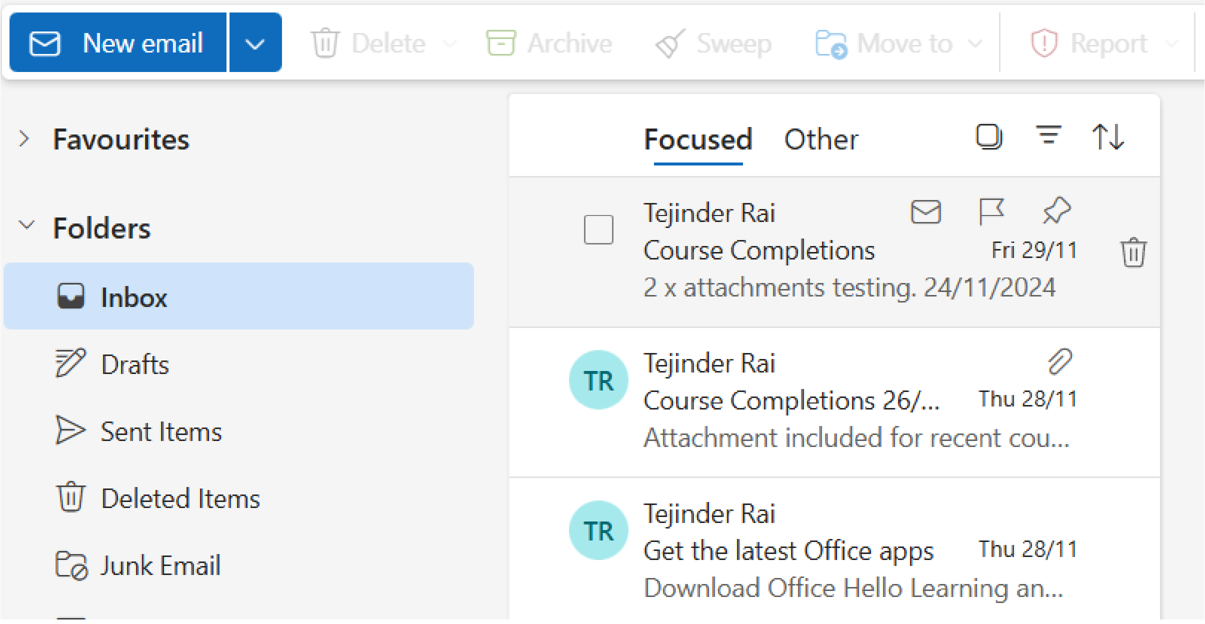
Application Runtime Process

The application process is described below.

1) The configuration is initialised
2) The messages are retrieved from the defined mailbox folder name
3) Each message is processed in the message collection and the emails with the matched string that are contained in the subject are processed
4) A console output of the message ID, received date, received from, and subject is displayed
5) Each attachment is processed and if the file is a file attachment, then the attachment is uploaded to the Azure file share specified in the Azure storage account connection string and file share name
6) The number of messages processed, and the number of attachments processed is displayed in the output of the console

Sample Output

The mailbox has two messages with the subject containing the search string “course completions”.

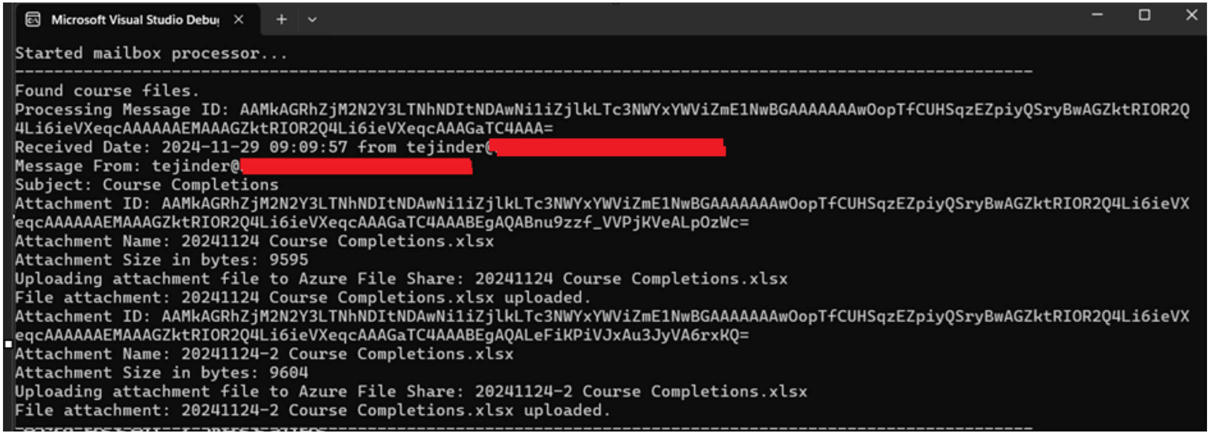


Tejinder Rai
Course Completions Fri 29/11
2 x attachments testing. 24/11/2024

Tejinder Rai
Course Completions 26/... Thu 28/11
Attachment included for recent cou...

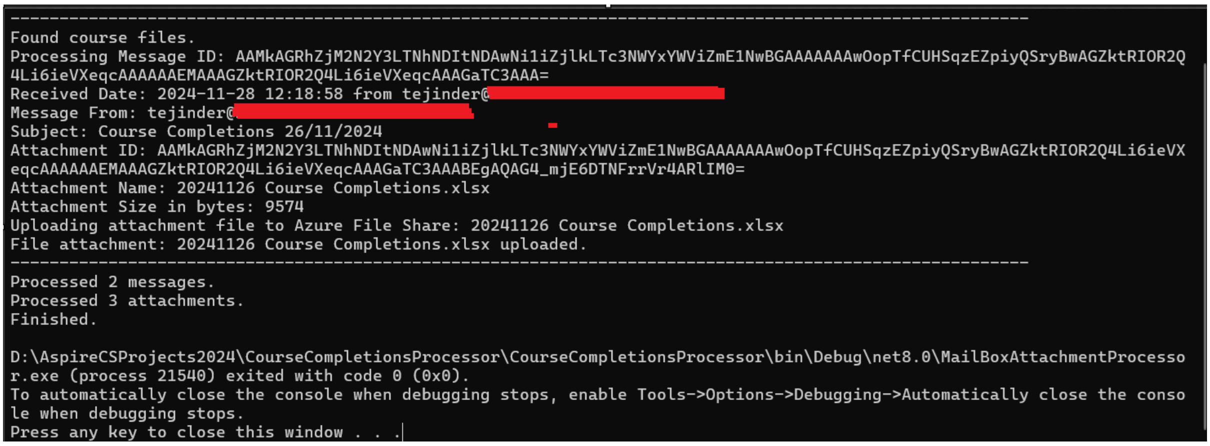
Tejinder Rai
Get the latest Office apps Thu 28/11
Download Office Hello Learning an...

The mailbox attachment processor is executed, and it displays the following output.



```
Started mailbox processor...
------------------------------------------------------------------------------------------------------------
Found course files.
Processing Message ID: AAMkAGRhZjM2N2Y3LTNhNDItNDAwNi1iZjlkLTc3NWYxYWViZmE1NwBGAAAAAAAwOopTfCUHSqzEZpiyQSryBwAGZktRIOR2Q
4Li6ieVXeqcAAAAAAEMAAAGZktRIOR2Q4Li6ieVXeqcAAAGaTC4AAA=
Received Date: 2024-11-29 09:09:57 from tejinder@[redacted]
Message From: tejinder@[redacted]
Subject: Course Completions
Attachment ID: AAMkAGRhZjM2N2Y3LTNhNDItNDAwNi1iZjlkLTc3NWYxYWViZmE1NwBGAAAAAAAwOopTfCUHSqzEZpiyQSryBwAGZktRIOR2Q4Li6ieVX
eqcAAAAAAEMAAAGZktRIOR2Q4Li6ieVXeqcAAAGaTC4AAABEgAQABnu9zzf_VVPjKVeALpOzWc=
Attachment Name: 20241124 Course Completions.xlsx
Attachment Size in bytes: 9595
Uploading attachment file to Azure File Share: 20241124 Course Completions.xlsx
File attachment: 20241124 Course Completions.xlsx uploaded.
Attachment ID: AAMkAGRhZjM2N2Y3LTNhNDItNDAwNi1iZjlkLTc3NWYxYWViZmE1NwBGAAAAAAAwOopTfCUHSqzEZpiyQSryBwAGZktRIOR2Q4Li6ieVX
eqcAAAAAAEMAAAGZktRIOR2Q4Li6ieVXeqcAAAGaTC4AAABEgAQALeFiKPiVJxAu3JyVA6rxKQ=
Attachment Name: 20241124-2 Course Completions.xlsx
Attachment Size in bytes: 9604
Uploading attachment file to Azure File Share: 20241124-2 Course Completions.xlsx
File attachment: 20241124-2 Course Completions.xlsx uploaded.
```

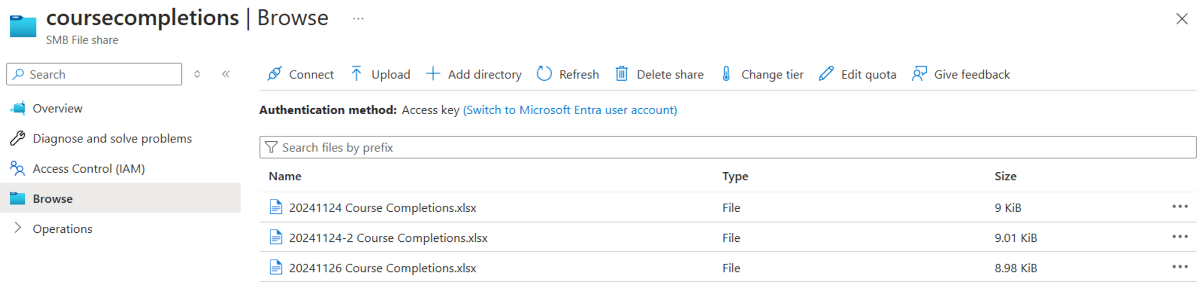
Two messages are processed, although three were seen in the previous email, but since the search string was not contained in the subject, only two messages were processed which were matched.



```
--------------------------------------------------------------------------------------------------------
Found course files.
Processing Message ID: AAMkAGRhZjM2N2Y3LTNhNDItNDAwNi1iZjlkLTc3NWYxYWViZmE1NwBGAAAAAAAwOopTfCUHSqzEZpiyQSryBwAGZktRIOR2Q
4Li6ieVXeqcAAAAAAEMAAAGZktRIOR2Q4Li6ieVXeqcAAAGaTC3AAA=
Received Date: 2024-11-28 12:18:58 from tejinder@[redacted]
Message From: tejinder@[redacted]
Subject: Course Completions 26/11/2024
Attachment ID: AAMkAGRhZjM2N2Y3LTNhNDItNDAwNi1iZjlkLTc3NWYxYWViZmE1NwBGAAAAAAAwOopTfCUHSqzEZpiyQSryBwAGZktRIOR2Q4Li6ieVX
eqcAAAAAAEMAAAGZktRIOR2Q4Li6ieVXeqcAAAGaTC3AAABEgAQAG4_mjE6DTNFrrVr4ARlIM0=
Attachment Name: 20241126 Course Completions.xlsx
Attachment Size in bytes: 9574
Uploading attachment file to Azure File Share: 20241126 Course Completions.xlsx
File attachment: 20241126 Course Completions.xlsx uploaded.
--------------------------------------------------------------------------------------------------------
Processed 2 messages.
Processed 3 attachments.
Finished.

D:\AspireCSProjects2024\CourseCompletionsProcessor\CourseCompletionsProcessor\bin\Debug\net8.0\MailBoxAttachmentProcesso
r.exe (process 21540) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

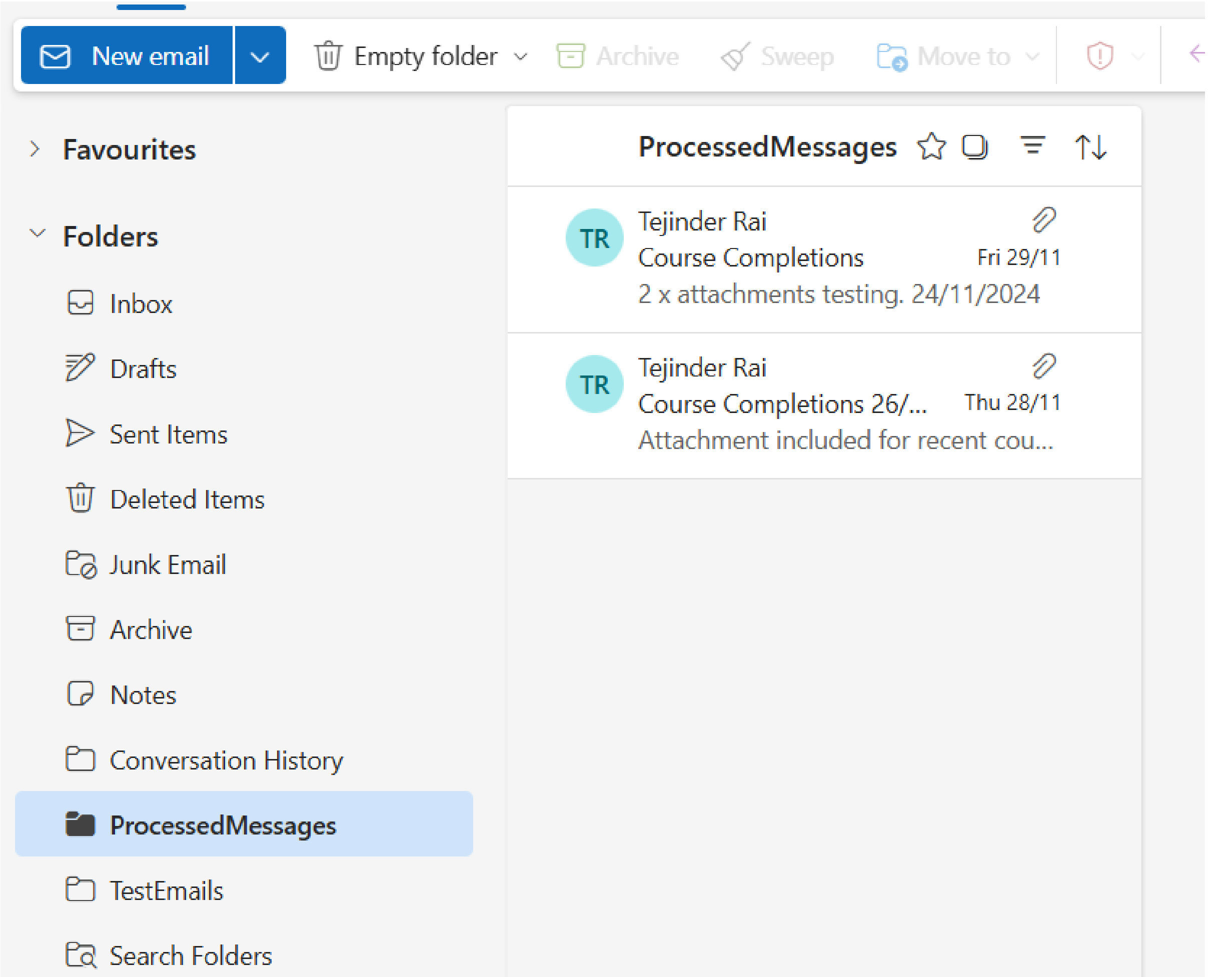
Three attachments in total were processed and uploaded to an Azure storage file share.



Authentication method: Access key (Switch to Microsoft Entra user account)

Name	Type	Size
20241124 Course Completions.xlsx	File	9 KiB
20241124-2 Course Completions.xlsx	File	9.01 KiB
20241126 Course Completions.xlsx	File	8.98 KiB

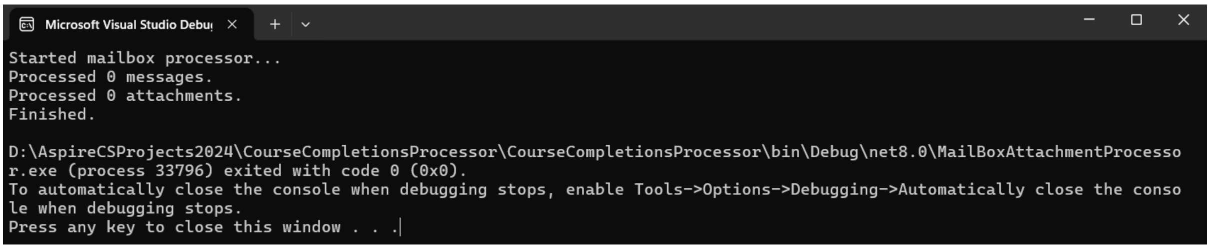
The email messages were moved to the ProcessedMessages folder, as defined in the application setting `ProcessedMessagesFolderName`.



Tejinder Rai
Course Completions Fri 29/11
2 x attachments testing. 24/11/2024

Tejinder Rai
Course Completions 26/... Thu 28/11
Attachment included for recent cou...

When the application is executed again, the output is shown below as there are no longer any matched messages to process.



```
Started mailbox processor...
Processed 0 messages.
Processed 0 attachments.
Finished.

D:\AspireCSProjects2024\CourseCompletionsProcessor\CourseCompletionsProcessor\bin\Debug\net8.0\MailBoxAttachmentProcesso
r.exe (process 33796) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Closing Thoughts

From a development point of view, using this method provides a simple solution. Other considerations:

- Store the storage account key in Azure Key Vault
- Store the application secret (if used(in Azure Key Vault
- The Azure resource hosting the application e.g. Function App, can have a managed identity and RBAC access can be provided to Azure Key Vault for the service principal (Azure Key Vault access policies are now deprecated)
- Environment settings can be stored in the hosting environment configuration rather than in the appsettings.json file.

References

Limiting application permissions to specific Exchange Online mailboxes - Microsoft Graph | Microsoft Learn