**Step 1: Create a New .NET Project**

1. **Open Visual Studio Code** and create a new folder for your project.
2. Select .NET 8.0 as the framework.
3. **Open a terminal** in VS Code and navigate to the newly created folder.
4. **Initialize a new .NET project** by running the following command:

dotnet new console -n S3Uploader

Upload the published files (S3Uploader.exe, appsettings.json, S3Uploader.deps.json, S3Uploader.runtimeconfig.json, and hostpolicy.dll) to an S3 bucket. Note: not all files are in the publish folder – run tests to see if we need to have these downloaded.

**Step 2: Set Up the Project**

1. **Navigate to the project directory**:

cd S3Uploader

**Add necessary packages**:

dotnet add package AWSSDK.S3

dotnet add package Microsoft.Extensions.Configuration

dotnet add package Microsoft.Extensions.Configuration.Json

**Update the Project File (S3Uploader.csproj)**

Ensure your project file looks like this: <Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>Exe</OutputType>

<TargetFramework>net8.0</TargetFramework>

<PublishSingleFile>true</PublishSingleFile>

<SelfContained>true</SelfContained>

<RuntimeIdentifier>win-x64</RuntimeIdentifier>

<PublishTrimmed>false</PublishTrimmed>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="AWSSDK.S3" Version="3.7.4.1" />

<PackageReference Include="Microsoft.Extensions.Configuration" Version="5.0.0" />

<PackageReference Include="Microsoft.Extensions.Configuration.Json" Version="5.0.0" />

</ItemGroup>

</Project>

**Step 4: Create appsettings.json**

Create an appsettings.json file in the project directory:

{

"AWS": {

"AccessKey": "your-access-key",

"SecretKey": "your-secret-key"

}

}

**Step 5: Implement the File Upload Logic**

Update the Program.cs file with the following code:

using System;

using System.IO;

using Amazon.S3;

using Amazon.S3.Transfer;

using Microsoft.Extensions.Configuration;

namespace S3Uploader

{

class Program

{

static void Main(string[] args)

{

if (args.Length != 1)

{

Console.WriteLine("Usage: S3Uploader <file-to-upload>");

return;

}

string filePath = args[0];

string bucketName = "authbillpro";

string keyName = Path.GetFileName(filePath);

var builder = new ConfigurationBuilder()

.SetBasePath(AppContext.BaseDirectory)

.AddJsonFile("appsettings.json", optional: false, reloadOnChange: true);

IConfiguration configuration = builder.Build();

string accessKey = configuration["AWS:AccessKey"];

string secretKey = configuration["AWS:SecretKey"];

try

{

var s3Client = new AmazonS3Client(accessKey, secretKey, Amazon.RegionEndpoint.USEast2);

var fileTransferUtility = new TransferUtility(s3Client);

fileTransferUtility.Upload(filePath, bucketName, keyName);

Console.WriteLine("File uploaded successfully.");

}

catch (AmazonS3Exception ex)

{

Console.WriteLine("Error encountered on server. Message:'{0}' when writing an object", ex.Message);

}

catch (Exception ex)

{

Console.WriteLine("Unknown encountered on server. Message:'{0}' when writing an object", ex.Message);

}

}

}

}

**Step 6: Publish the Project**

1. **Build and publish the project**:

 Ensure the .NET project is published as a single file, self-contained executable.

1.  Include necessary files like appsettings.json for configuration.

dotnet publish -c Release -r win-x64 --self-contained

**Upload the published files to AWS S3**:

**Step 7: VBA Script for File Upload**

**VBA Code**:

Private Declare PtrSafe Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" ( \_

ByVal hwnd As LongPtr, ByVal lpOperation As String, ByVal lpFile As String, \_

ByVal lpParameters As String, ByVal lpDirectory As String, ByVal nShowCmd As Long) As LongPtr

Private Sub btnUploadFile\_Click()

Dim tempFolderPath As String

tempFolderPath = Environ("TEMP") & "\S3UploadTemp\"

' Ensure the temp directory exists

If Dir(tempFolderPath, vbDirectory) = "" Then

MkDir tempFolderPath

End If

'  Download the necessary files from S3 to a local temporary directory.

 Prompt the user to select a file for upload.

 Execute the .NET executable with the selected file as a parameter.

 Handle and log any errors that occur during the process.

DownloadFile "https://authbillpro.s3.us-east-2.amazonaws.com/publish/S3Uploader.exe", tempFolderPath & "S3Uploader.exe"

DownloadFile "https://authbillpro.s3.us-east-2.amazonaws.com/publish/appsettings.json", tempFolderPath & "appsettings.json"

DownloadFile "https://authbillpro.s3.us-east-2.amazonaws.com/publish/S3Uploader.deps.json", tempFolderPath & "S3Uploader.deps.json"

DownloadFile "https://authbillpro.s3.us-east-2.amazonaws.com/publish/S3Uploader.runtimeconfig.json", tempFolderPath & "S3Uploader.runtimeconfig.json"

DownloadFile "https://authbillpro.s3.us-east-2.amazonaws.com/publish/hostpolicy.dll", tempFolderPath & "hostpolicy.dll"

' Allow the user to select a file to upload

Dim fd As fileDialog

Set fd = Application.fileDialog(msoFileDialogFilePicker)

Dim selectedFilePath As String

With fd

.Title = "Select a File to Upload"

.Filters.Add "All Files", "\*.\*"

If .Show = -1 Then

selectedFilePath = .SelectedItems(1)

Else

Exit Sub ' User cancelled

End If

End With

' Prepare the command to execute

Dim exeFilePath As String

exeFilePath = tempFolderPath & "S3Uploader.exe"

Dim command As String

command = """" & exeFilePath & """ """ & selectedFilePath & """"

' Log the process

Dim errorLogPath As String

errorLogPath = tempFolderPath & "upload\_error\_log.txt"

Open errorLogPath For Output As #1

Print #1, "Starting file upload process: " & Now

Print #1, "Command to execute: " & command

' Execute the command

Dim result As LongPtr

result = ShellExecute(0, "open", exeFilePath, selectedFilePath, vbNullString, 1)

Print #1, "Exit code: " & result

' Check the result

If result <= 32 Then

Print #1, "File upload failed with error code: " & result

MsgBox "File upload failed with error code: " & result, vbCritical

Else

Print #1, "File uploaded successfully."

MsgBox "File uploaded successfully.", vbInformation

End If

Close #1

End Sub

Private Sub DownloadFile(url As String, destPath As String)

Dim http As Object

Set http = CreateObject("MSXML2.XMLHTTP")

http.Open "GET", url, False

http.send

If http.status = 200 Then

Dim stream As Object

Set stream = CreateObject("ADODB.Stream")

stream.Type = 1 ' adTypeBinary

stream.Open

stream.Write http.responseBody

stream.SaveToFile destPath, 2 ' adSaveCreateOverWrite

stream.Close

Else

MsgBox "Error downloading file: " & url & vbCrLf & "Status: " & http.status & " - " & http.statusText, vbCritical

End If

End Sub

Note: make sure file name has no spaces.