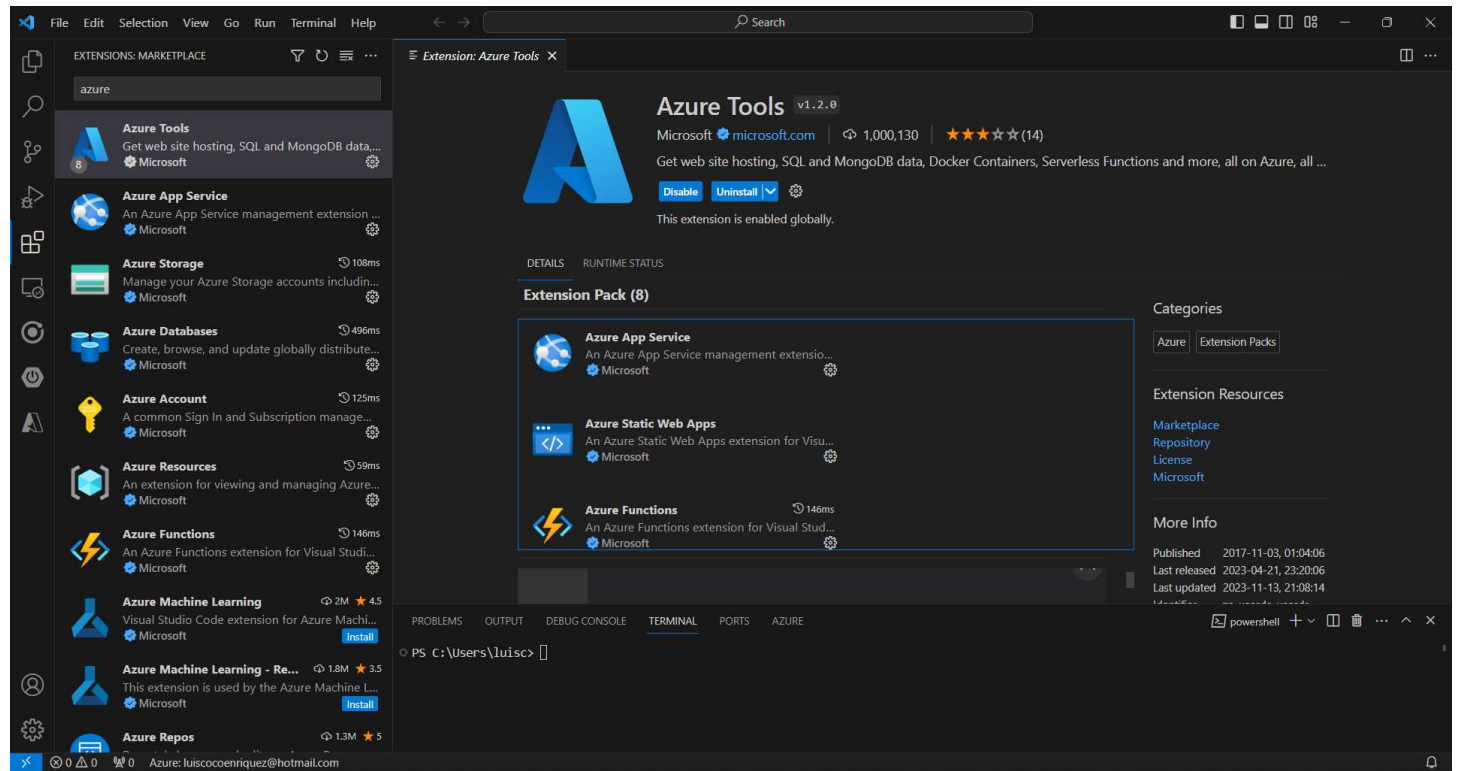


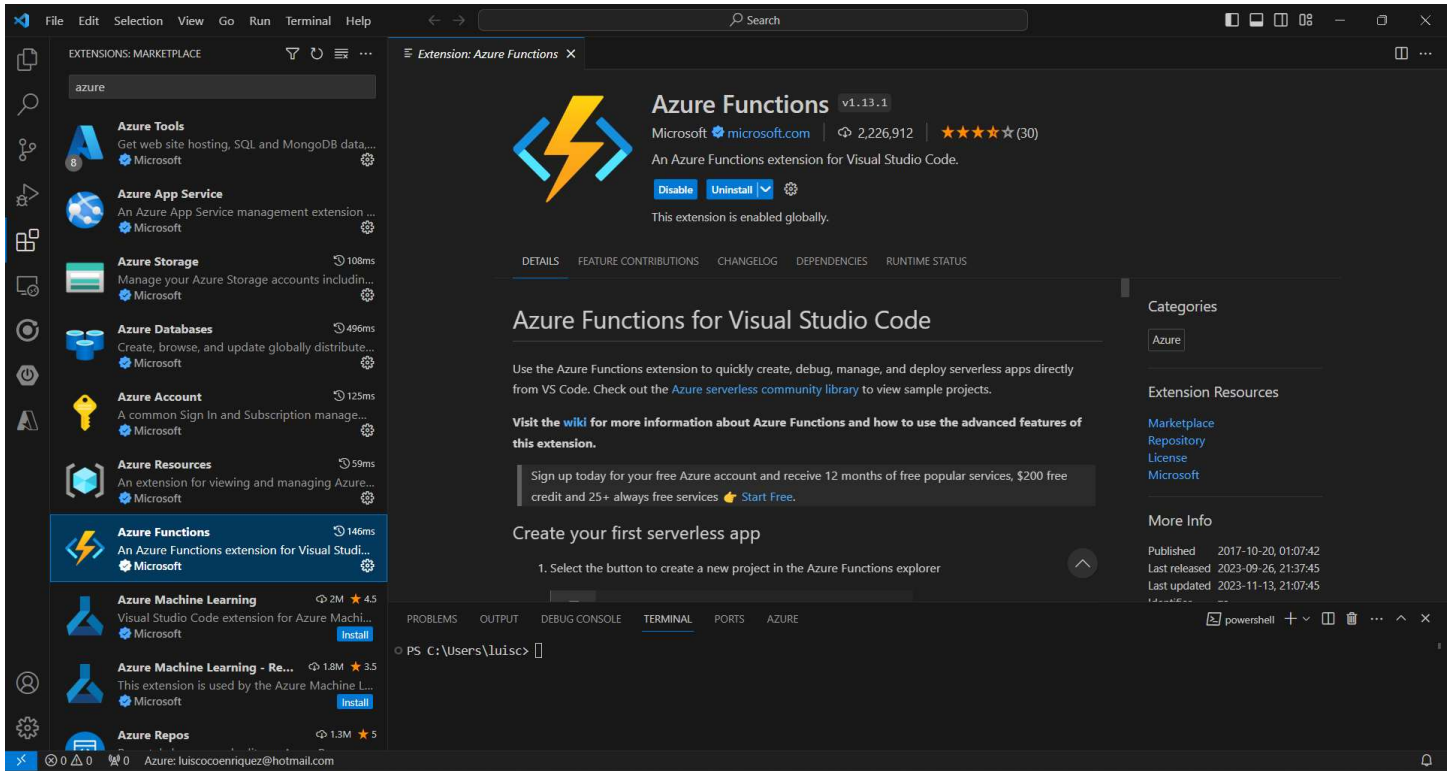
How to create an Azure ResourceGroup from an Azure Function (HttpTriggered)

1. Run VSCode and install the extensions

First we have to install "Azure Tools" extension.

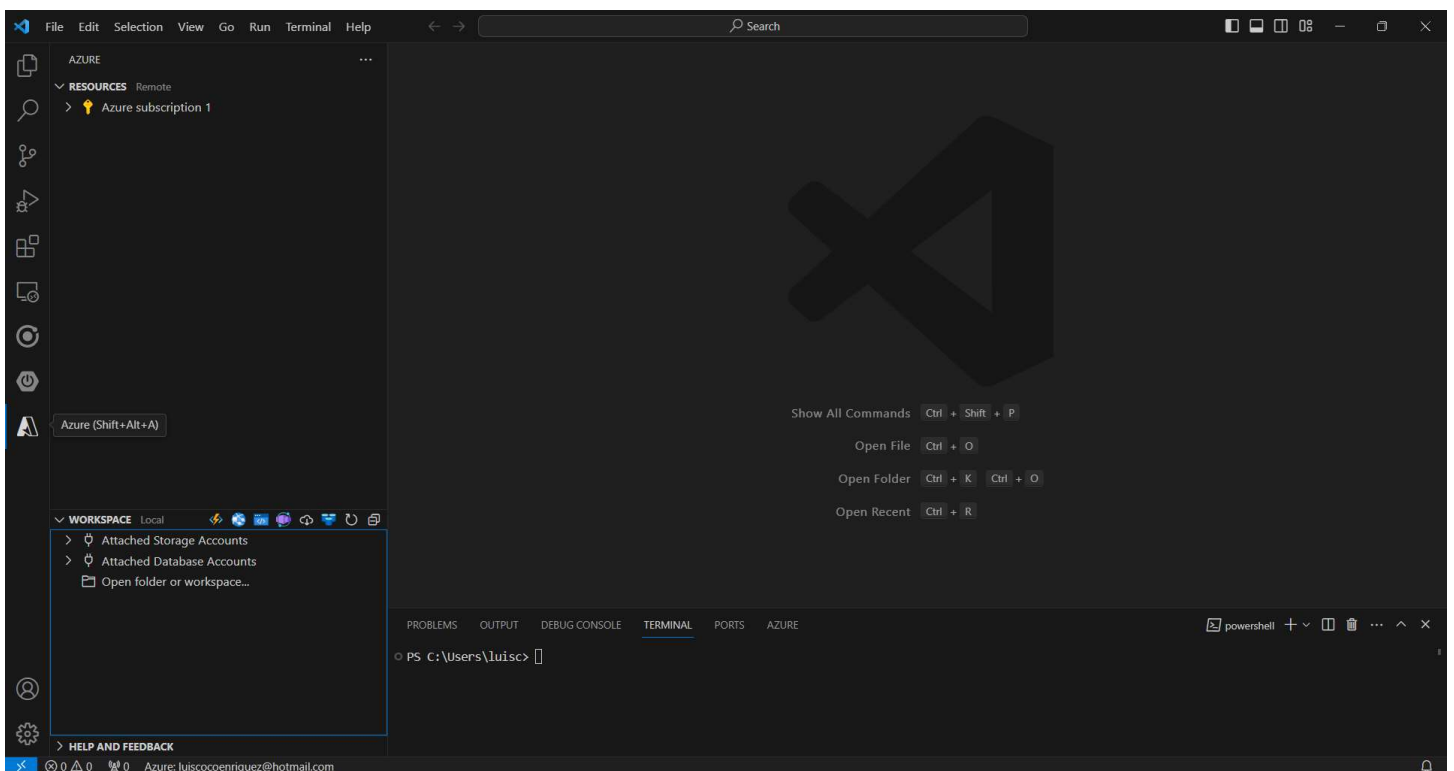


Then we install the "Azure Functions" extension.

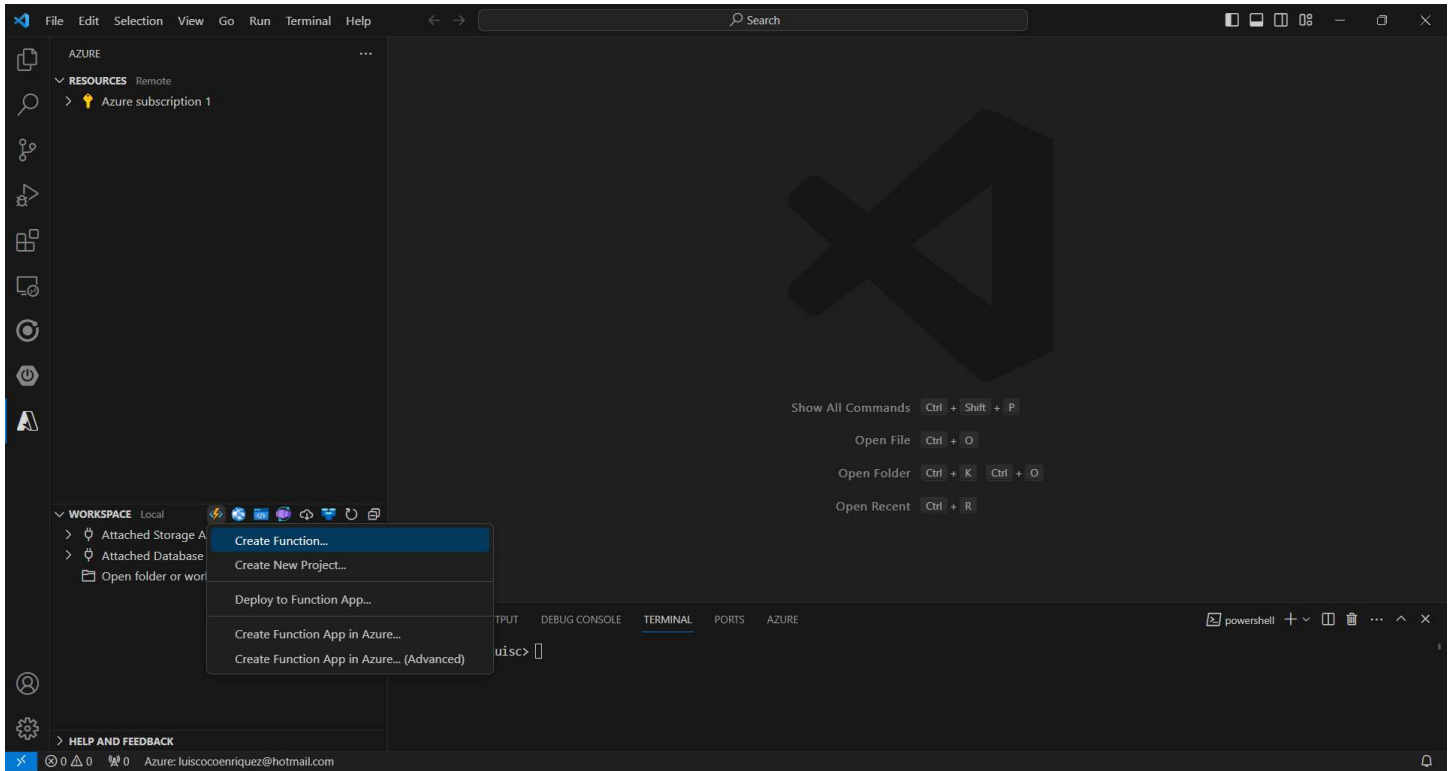


2. Create a new Azure Function with VSCode

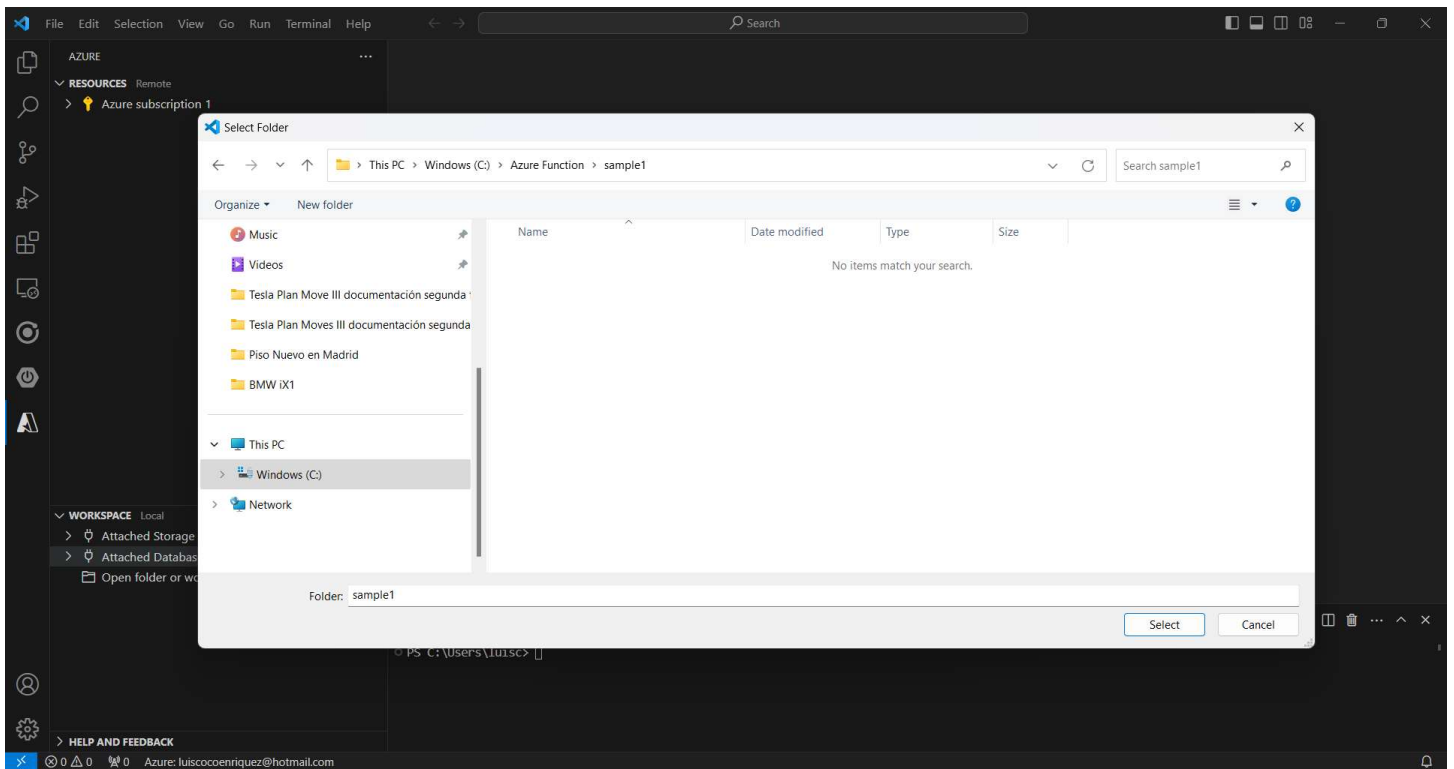
Click on the "Azure Tools" extension icon in the left menu.



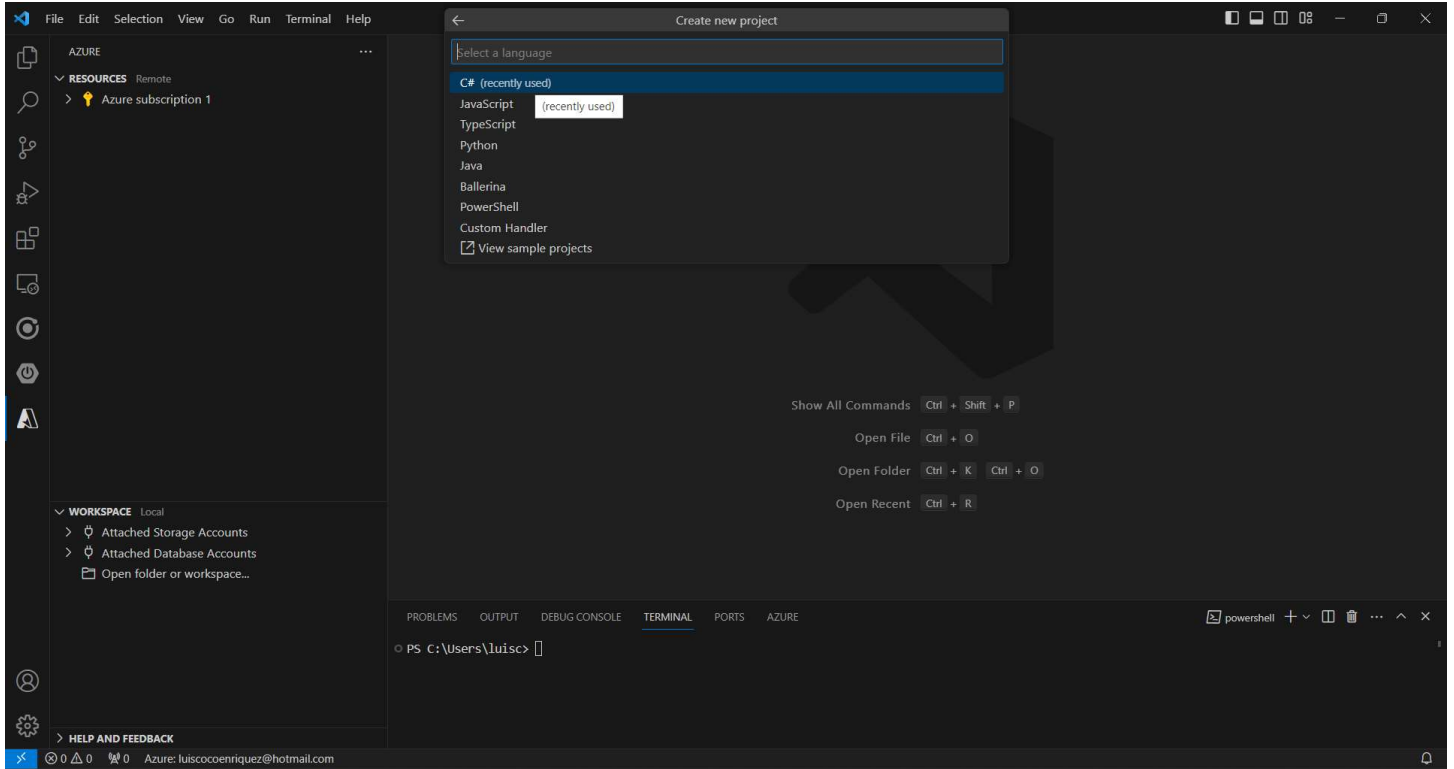
Select the "Create Function..." option in the menu



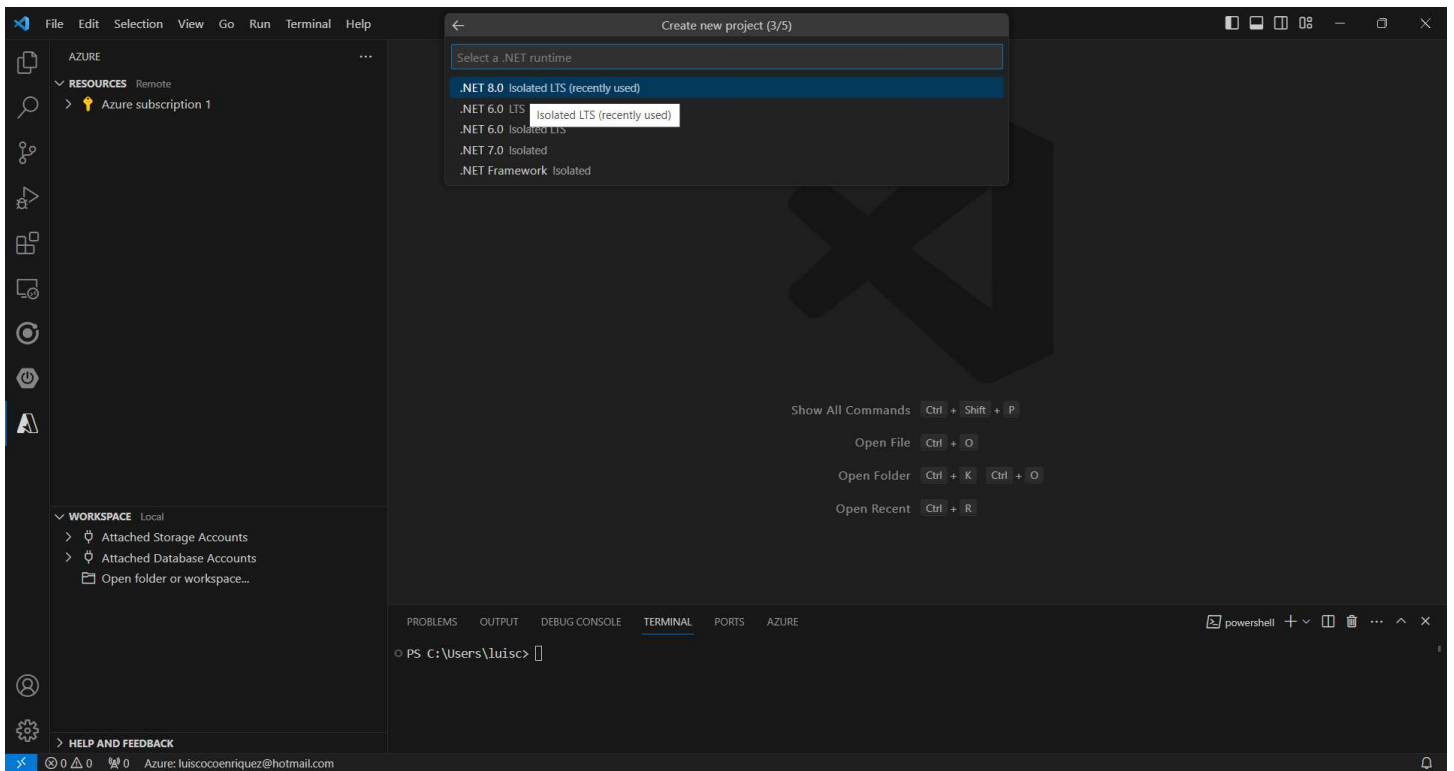
Select the folder `"/Azure Function/sample1/"` where to place the new Azure Function:



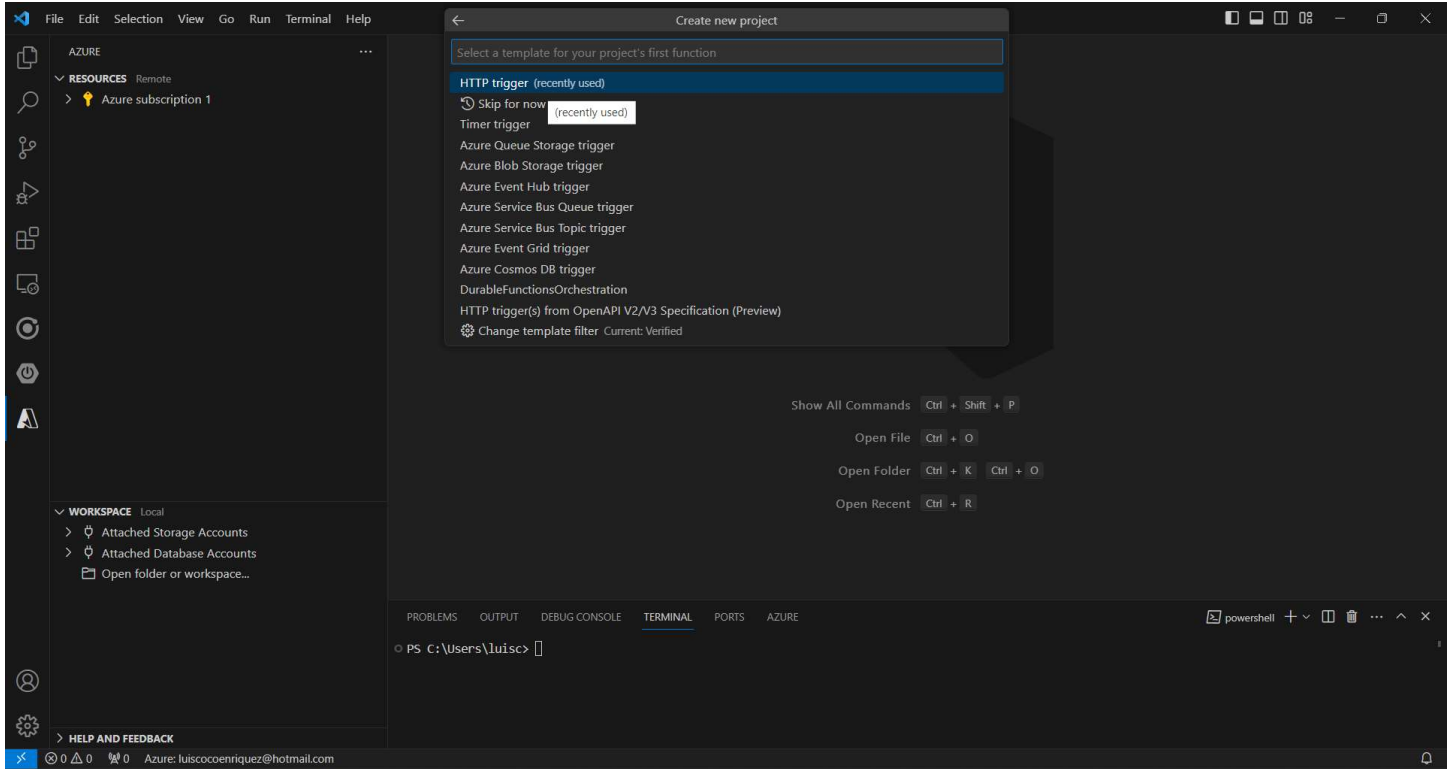
Select the Azure Function language, in this case we select **C#**:



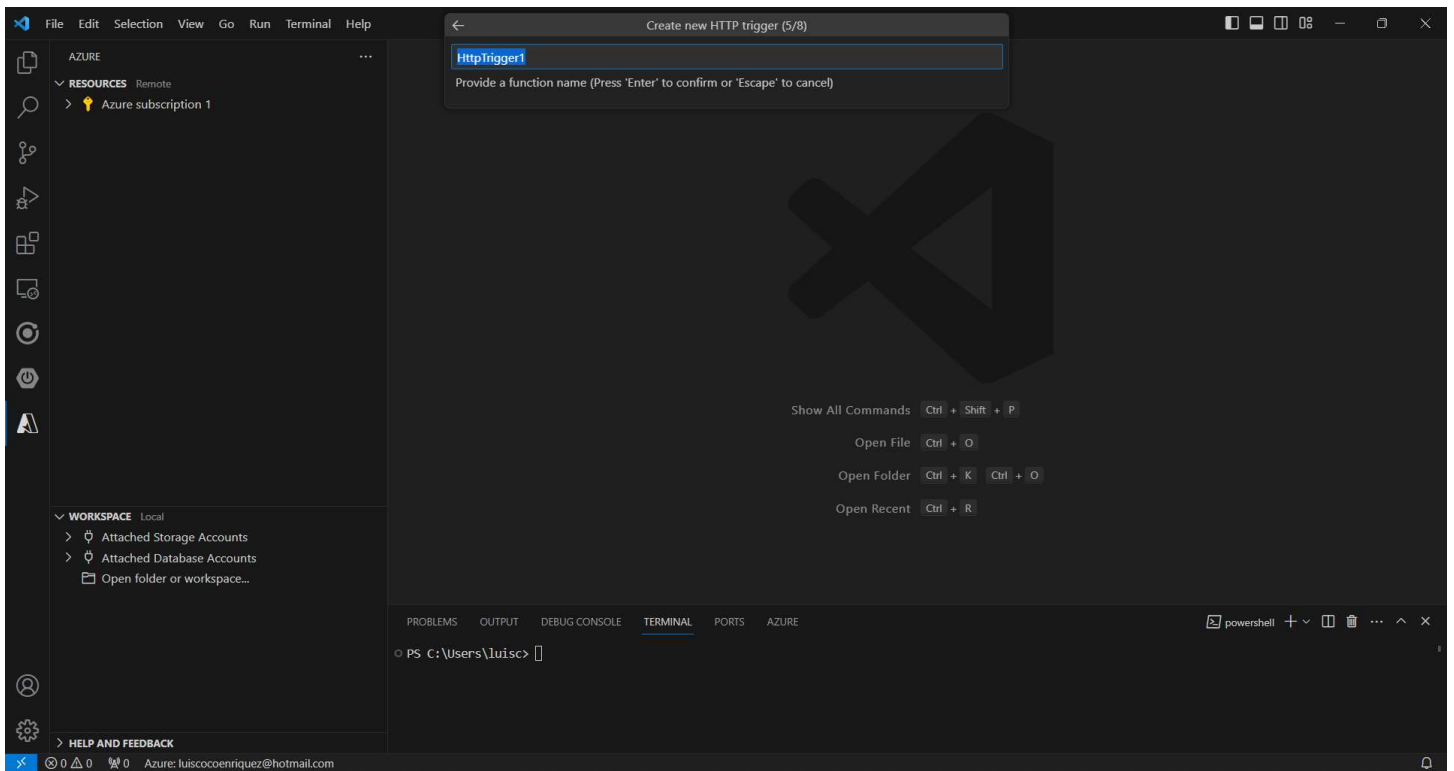
Select the Azure Function runtime libraries, we select .NET 8:



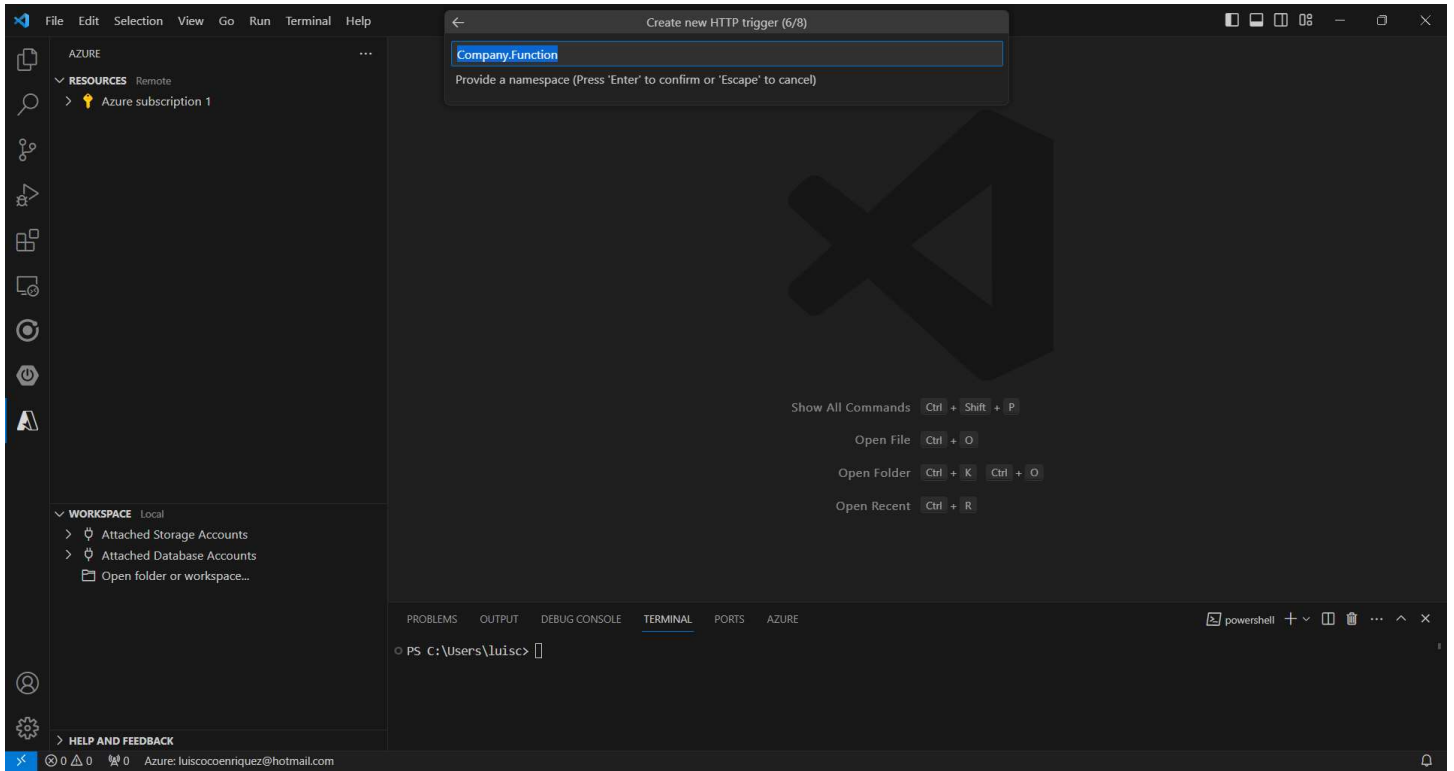
The we select the Azure Function trigger, we select HTTP triggered



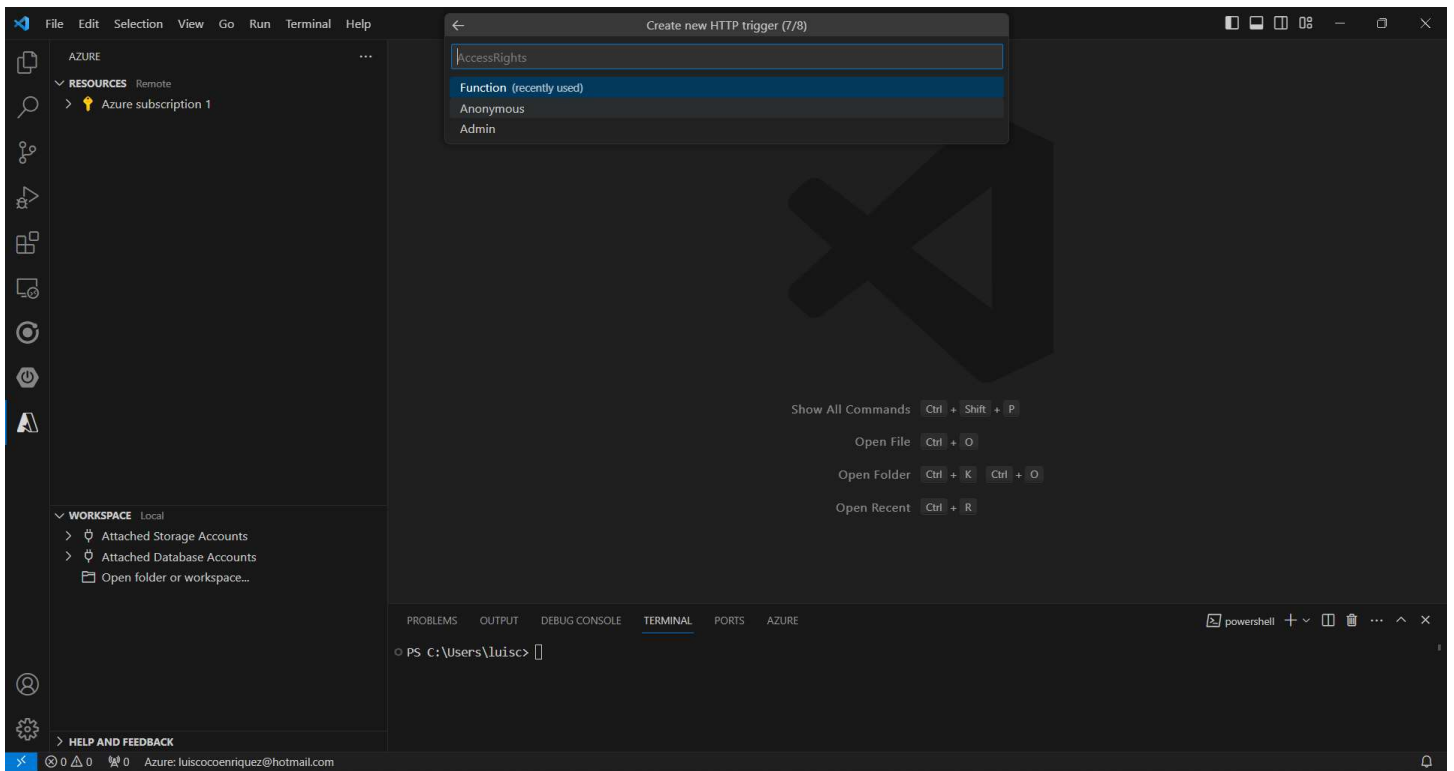
Now we provide the Azure Function name:



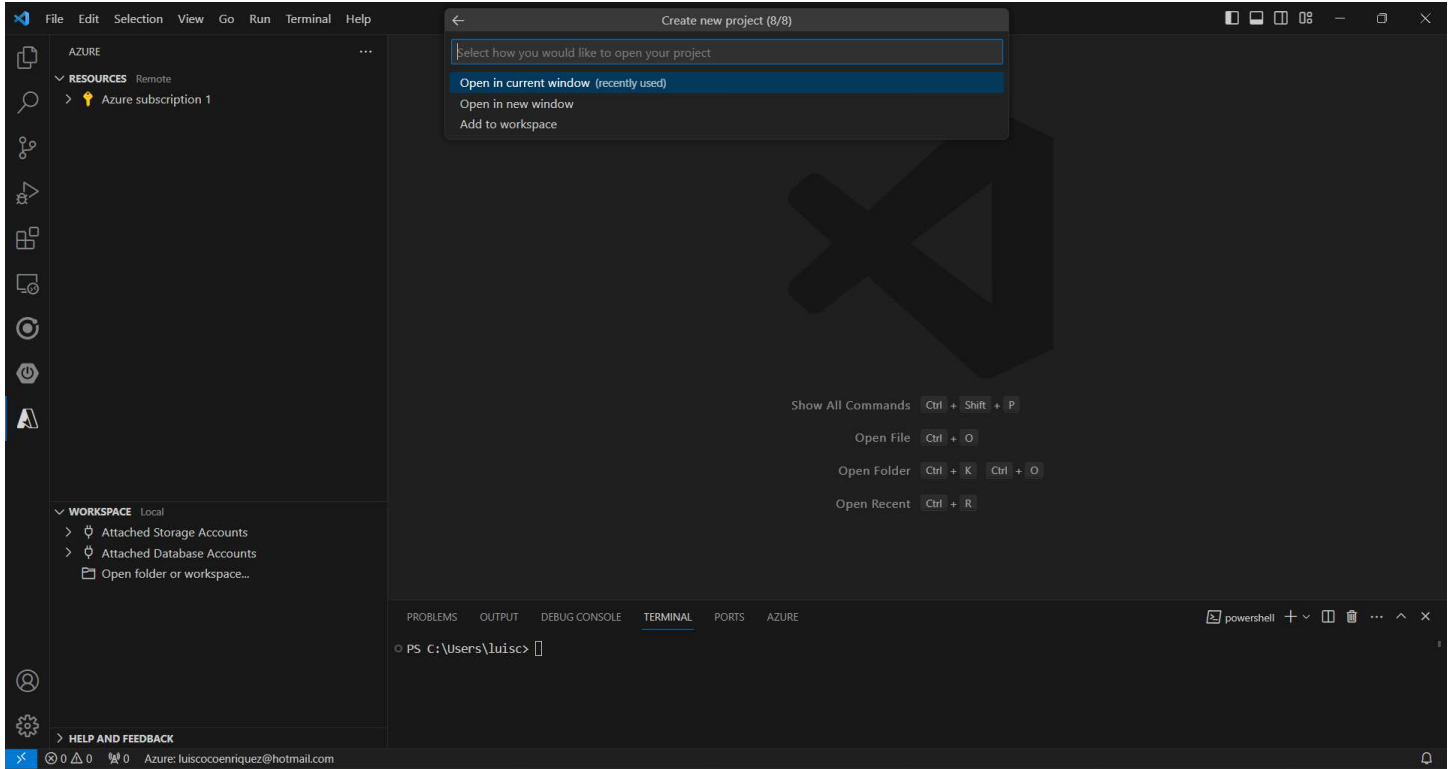
We also set the Azure Function namespace:



Now we grant the access rights "Function":



Finally, we open in the current window the new Azure Function source code:

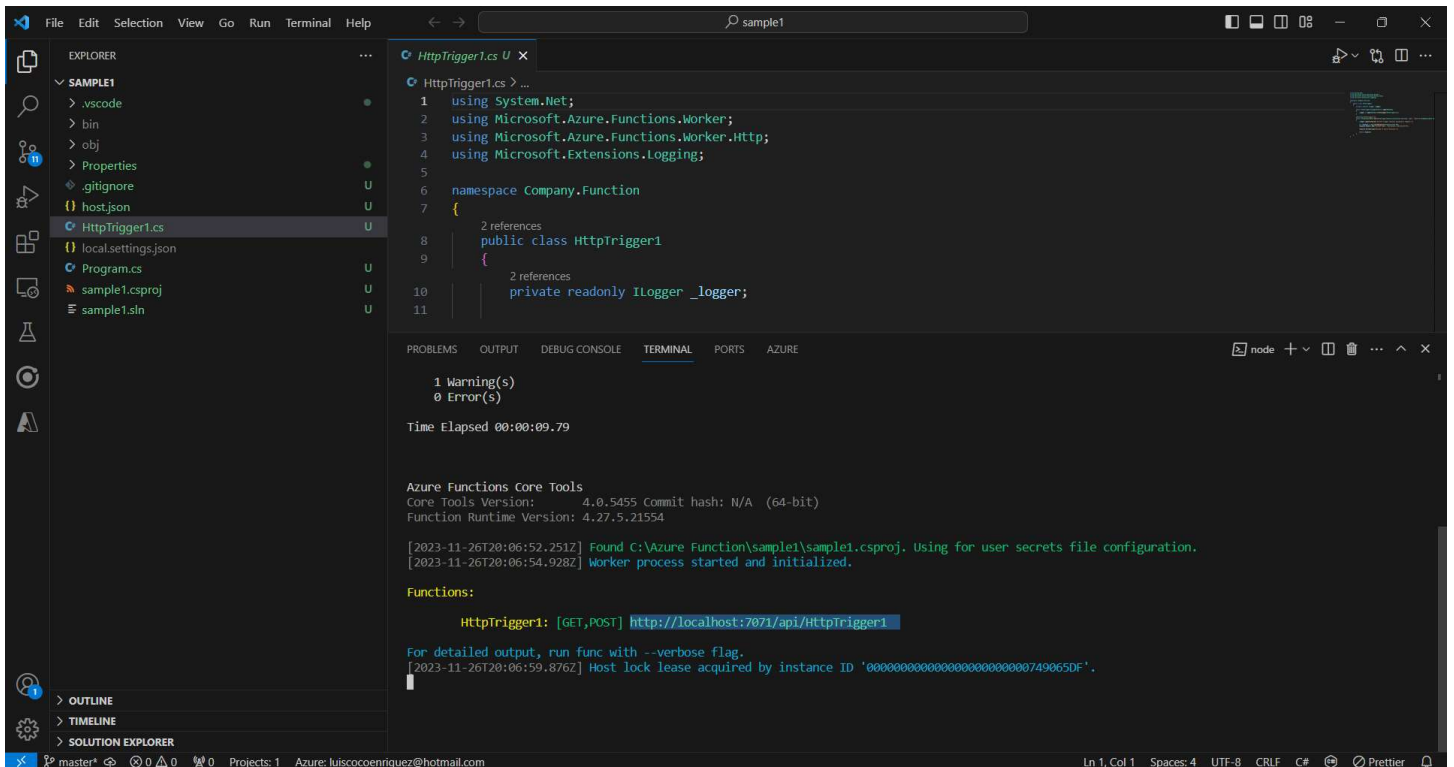


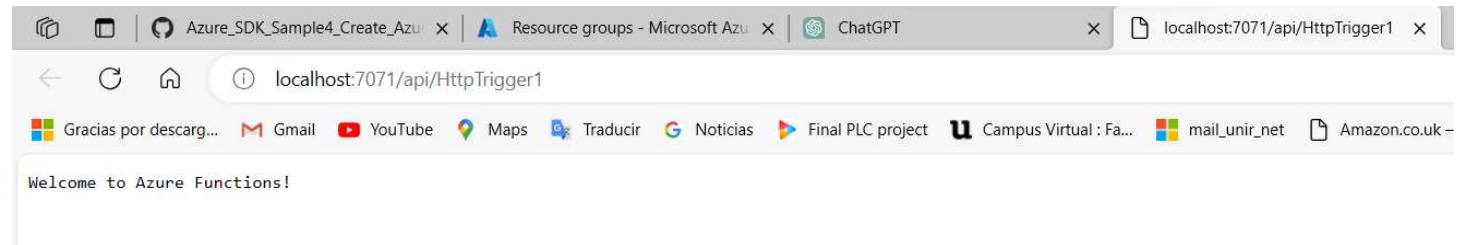
For running the function we type the command:

```
func start
```

To access the Azure Function endpoint we navigate to this URL:

<http://localhost:7071/api/HttpTrigger1>





Press **Ctrl+C** for stopping the application.

3. Load the Azure SDK for .NET libraries

In Visual Studio Code, you can use the terminal to install NuGet packages for your C# application.

The command to install the **Azure SDK libraries** would be. We type these commands in the terminal window in VSCode:

```
dotnet add package Azure.Identity
dotnet add package Azure.ResourceManager.Resources
dotnet add package Azure.ResourceManager
```

Now we reference the libraries in the C# source code:

```
...
using Azure.Core;
using Azure.Identity;
using Azure.ResourceManager.Resources;
using Azure.ResourceManager;
using Azure;
...
```

4. We input the C# source code for creating the Azure Resource Group with Azure SDK for .NET

This is the code for creating the new Azure Resource Group:

```
ArmClient armClient = new ArmClient(new DefaultAzureCredential());
SubscriptionResource subscription = await armClient.GetDefaultSubscriptionAsync();
string rgName = "myNewRgName";
AzureLocation location = AzureLocation.WestEurope;
ArmOperation<ResourceGroupResource> operation = await subscription.GetResourceGroups().CreateOr
ResourceGroupResource resourceGroup = operation.Value;
Console.WriteLine(resourceGroup.Data.Name);
```


This code is written in C# and uses the Azure SDK for .NET to interact with Azure Resource Manager (ARM). Let me break it down for you:

4.1. ArmClient Initialization:

```
ArmClient armClient = new ArmClient(new DefaultAzureCredential());
```

Here, an instance of ArmClient is created using the DefaultAzureCredential().

DefaultAzureCredential is part of the Azure Identity library and is used for automatically managing Azure Active Directory (AAD) authentication.

It tries to use various methods for authentication, such as environment variables, managed identity, or interactive login.

4.2. Get Default Subscription:

```
SubscriptionResource subscription = await armClient.GetDefaultSubscriptionAsync();
```

It retrieves the default subscription for the authenticated user.

The await keyword indicates that this operation is asynchronous.

4.3. Resource Group Information:

```
string rgName = "myRgNameLCESEGUNDO";  
AzureLocation location = AzureLocation.WestEurope;
```

Sets the name and location of the Azure Resource Group that you want to create or update.

4.4. Create or Update Resource Group:

```
ArmOperation<ResourceGroupResource> operation = await subscription.GetResourceGroups().CreateOr
```



Initiates an asynchronous operation to create or update the specified resource group.

It uses the **CreateOrUpdateAsync** method on the ResourceGroups property of the subscription.

The **WaitUntilCompleted** parameter indicates that the code should wait until the operation is completed.

4.5. Get Resource Group Information:

```
ResourceGroupResource resourceGroup = operation.Value;
```

Retrieves the result of the asynchronous operation.

The Value property contains the actual **ResourceGroupResource** object.

4.6. Print Resource Group Name:

```
Console.WriteLine(resourceGroup.Data.Name);
```

Prints the resourcegroup name in the console.

In summary, this code authenticates the user using Azure Identity, gets the default subscription, creates or updates a resource group with a specified name and location, and then prints the name of the resource group to the console.

5. This is the whole Azure Function C# source code.

```
using System.Net;
using Azure.Core;
using Azure.Identity;
using Azure.ResourceManager.Resources;
using Azure.ResourceManager;
using Azure;
using Microsoft.Azure.Functions.Worker;
using Microsoft.Azure.Functions.Worker.Http;
using Microsoft.Extensions.Logging;

namespace Company.Function
{
    public class HttpTrigger1
    {
        private readonly ILogger _logger;

        public HttpTrigger1(ILoggerFactory loggerFactory)
        {
            _logger = loggerFactory.CreateLogger<HttpTrigger1>();
        }

        [Function("HttpTrigger1")]
        public async Task<HttpResponseBody> RunAsync([HttpTrigger(AuthorizationLevel.Function,
        {
            _logger.LogInformation("C# HTTP trigger function processed a request.");

            var response = req.CreateResponse(HttpStatusCode.OK);
            response.Headers.Add("Content-Type", "text/plain; charset=utf-8");
        }
    }
}
```

```
response.WriteString("Welcome to Azure Functions!");

ArmClient armClient = new ArmClient(new DefaultAzureCredential());
SubscriptionResource subscription = await armClient.GetDefaultSubscriptionAsync();

string rgName = "myNewRgName";
AzureLocation location = AzureLocation.WestEurope;
ArmOperation<ResourceGroupResource> operation = await subscription.GetResourceGroupResource(location, rgName);
ResourceGroupResource resourceGroup = operation.Value;
Console.WriteLine(resourceGroup.Data.Name);

return response;
}
}
```

IMPORTANT NOTE: please pay attention and see the Azure Function call with "RunAsync" method.

We used the "async" and the "Task" keywords:

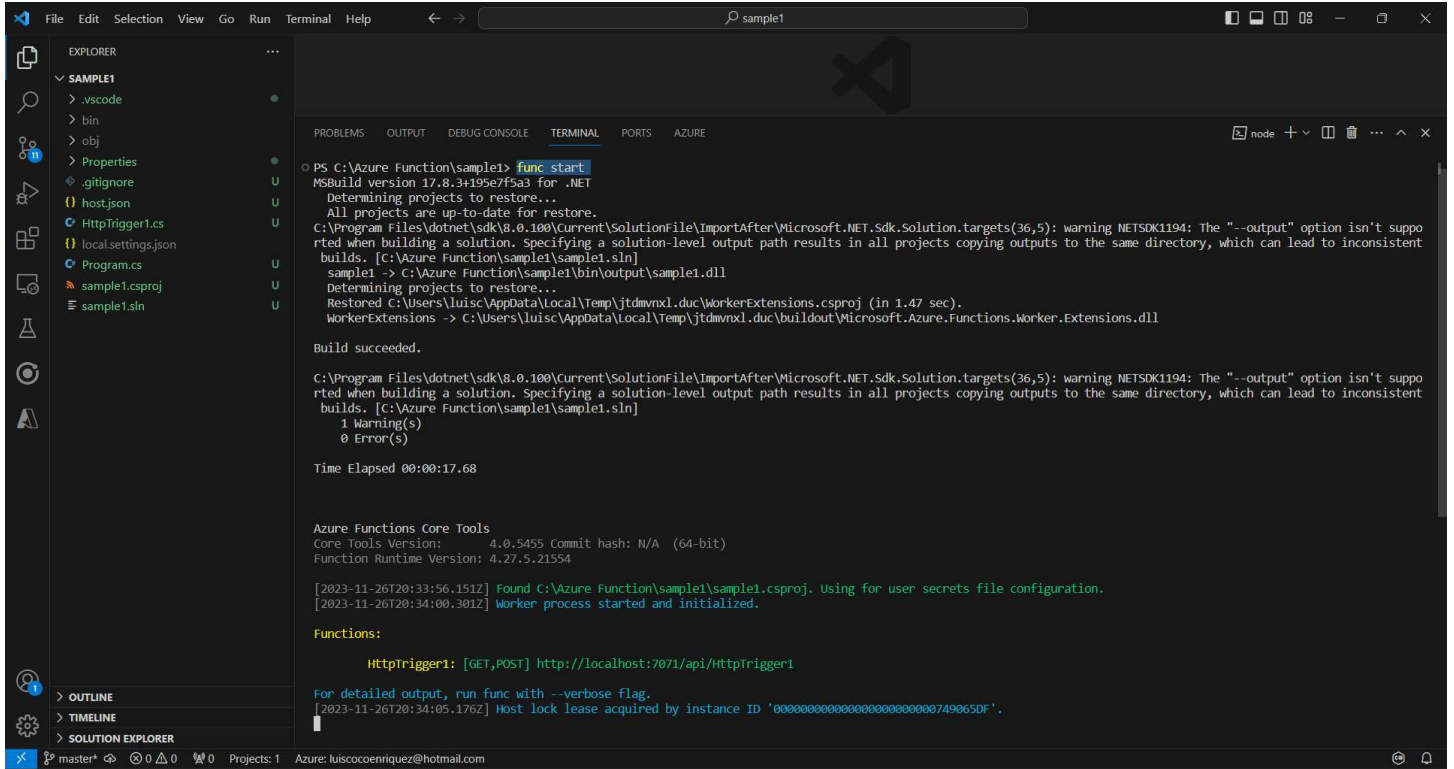
```
[Function("HttpTrigger1")]
public async Task<HttpResponseBody> RunAsync([HttpTrigger(AuthorizationLevel.Function,
```

6. We run the Azure Function

We execute the Azure Function with the command:

```
func start
```

See the output after running the Azure Function:



```
PS C:\Azure Function\sample1> func start
MSBuild version 17.8.3+195e7f5a3 for .NET
Determining projects to restore...
All projects are up-to-date for restore.
C:\Program Files\dotnet\sdk\8.0.100\Current\SolutionFile\ImportAfter\Microsoft.NET.Sdk.Solution.targets(36,5): warning NETSDK1194: The "--output" option isn't supported when building a solution. Specifying a solution-level output path results in all projects copying outputs to the same directory, which can lead to inconsistent builds. [C:\Azure Function\sample1\sample1.sln]
sample1 -> C:\Azure Function\sample1\bin\output\sample1.dll
Determining projects to restore...
Restored C:\Users\luisc\AppData\Local\Temp\jtdmvmxl.duc\WorkerExtensions.csproj (in 1.47 sec).
WorkerExtensions -> C:\Users\luisc\AppData\Local\Temp\jtdmvmxl.duc\buildout\Microsoft.Azure.Functions.Worker.Extensions.dll

Build succeeded.
Time Elapsed 00:00:17.68

Azure Functions Core Tools
Core Tools Version: 4.0.5455 Commit hash: N/A (64-bit)
Function Runtime Version: 4.27.5.21554

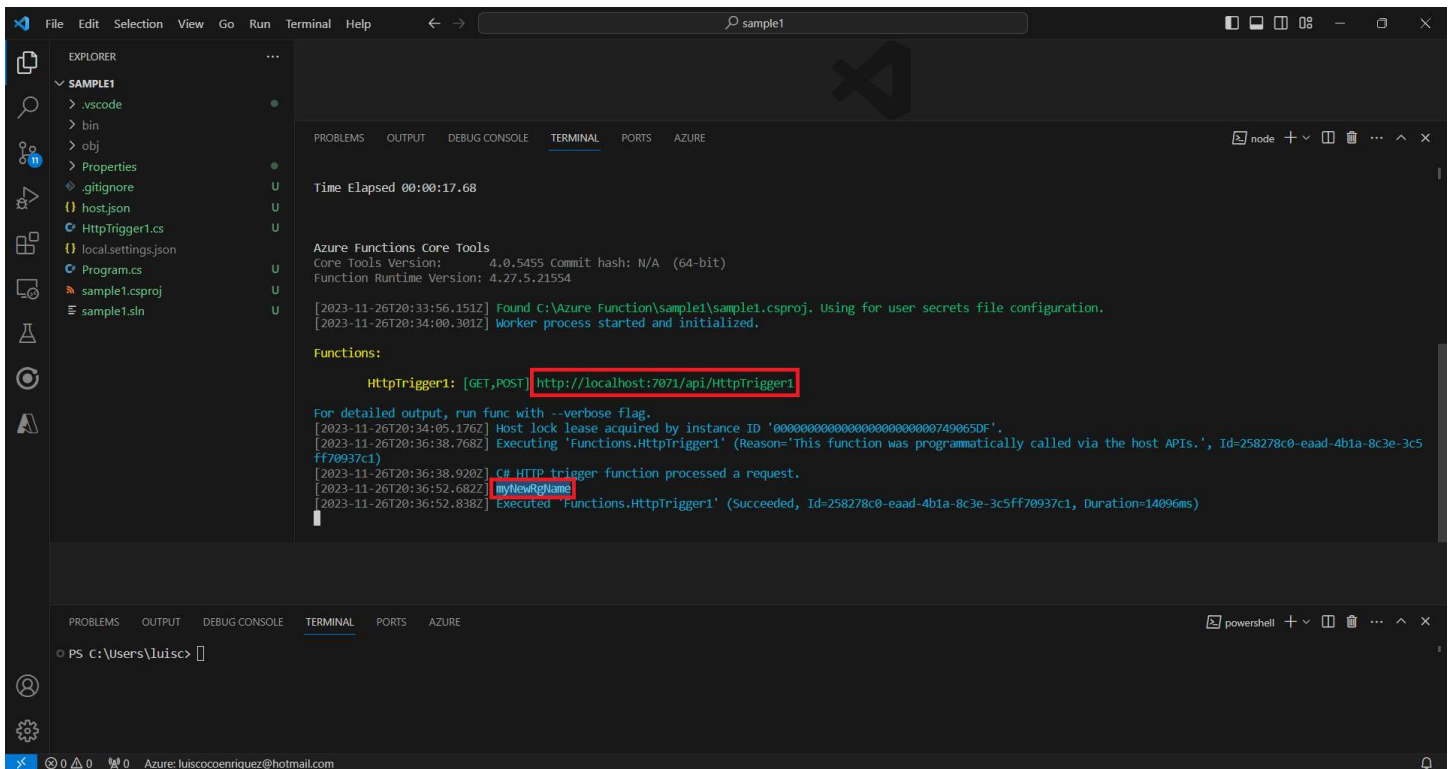
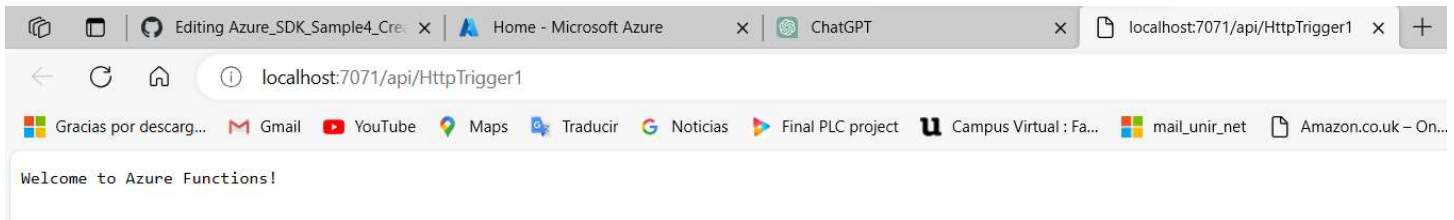
[2023-11-26T20:33:56.151Z] Found C:\Azure Function\sample1\sample1.csproj. Using for user secrets file configuration.
[2023-11-26T20:34:00.301Z] Worker process started and initialized.

Functions:

    HttpTrigger1: [GET,POST] http://localhost:7071/api/HttpTrigger1

For detailed output, run func with --verbose flag.
[2023-11-26T20:34:05.176Z] Host lock lease acquired by instance ID '000000000000000000000000749065DF'.
```

Now we navigate to the Azure Function URL then we can check the new Azure Resource Group was created.



```
Time Elapsed 00:00:17.68

Azure Functions Core Tools
Core Tools Version: 4.0.5455 Commit hash: N/A (64-bit)
Function Runtime Version: 4.27.5.21554

[2023-11-26T20:33:56.151Z] Found C:\Azure Function\sample1\sample1.csproj. Using for user secrets file configuration.
[2023-11-26T20:34:00.301Z] Worker process started and initialized.

Functions:

    HttpTrigger1: [GET,POST] http://localhost:7071/api/HttpTrigger1

For detailed output, run func with --verbose flag.
[2023-11-26T20:34:05.176Z] Host lock lease acquired by instance ID '000000000000000000000000749065DF'.
[2023-11-26T20:36:38.768Z] Executing 'Functions.HttpTrigger1' (Reason='This function was programmatically called via the host APIs.', Id=258278c0-eaad-4b1a-8c3e-3c5ff70937c1)
[2023-11-26T20:36:38.920Z] C# HTTP trigger function processed a request.
[2023-11-26T20:36:52.682Z] myNewRgName
[2023-11-26T20:36:52.838Z] Executed 'Functions.HttpTrigger1' (Succeeded, Id=258278c0-eaad-4b1a-8c3e-3c5ff70937c1, Duration=14096ms)
```

```
PS C:\Users\luisc>
```

7. For checking we created the new ResourceGroup in Azure

We login in Azure

We select "Resource Groups" and we check the new resource group named "" was created:

