Sending Message Via Application

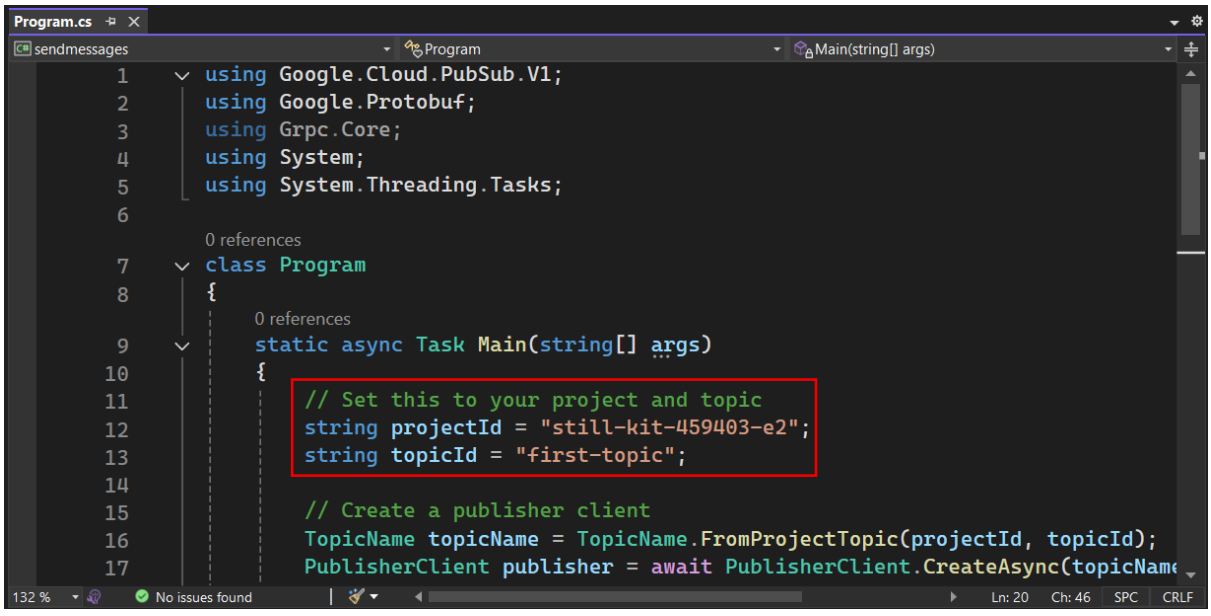
1. Now we will use a .NET application to send messages to Pub/Sub GCP.
2. For that you will find a zip file named send message on the GitHub repository. Download it and unzip the file. Then open the solution file in Visual Studio 2022.
3. Inside your Visual Studio, open **Tools > NuGet Package Manager > Package Manager Console**.
4. Then run the following command to install the packages for Google cloud Pub/Sub.

   Install-Package Google.Cloud.PubSub.V1

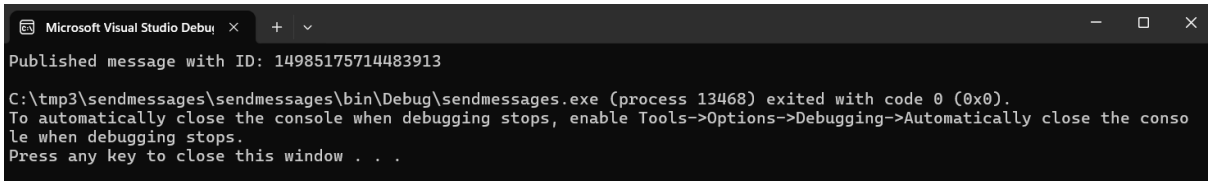
5. Also, Google CLI should be installed on your local machine. Then open the CLI and run the command given below. This command will help you to log in with your Gmail account that you are using on GCP.

   gcloud auth application-default login

6. Once you have logged in, come back or open the solution file for the .NET application.
7. Here, you have to open the Program.cs file and change the project ID and topic ID from GCP. Then save your file.

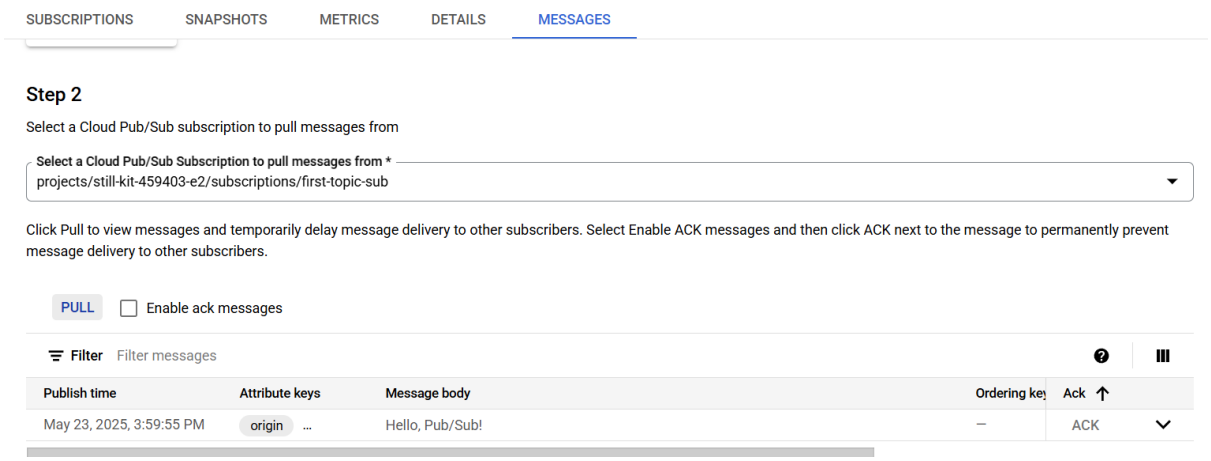


8. After that, just run your application. If everything works fine, then you will get the published message ID.

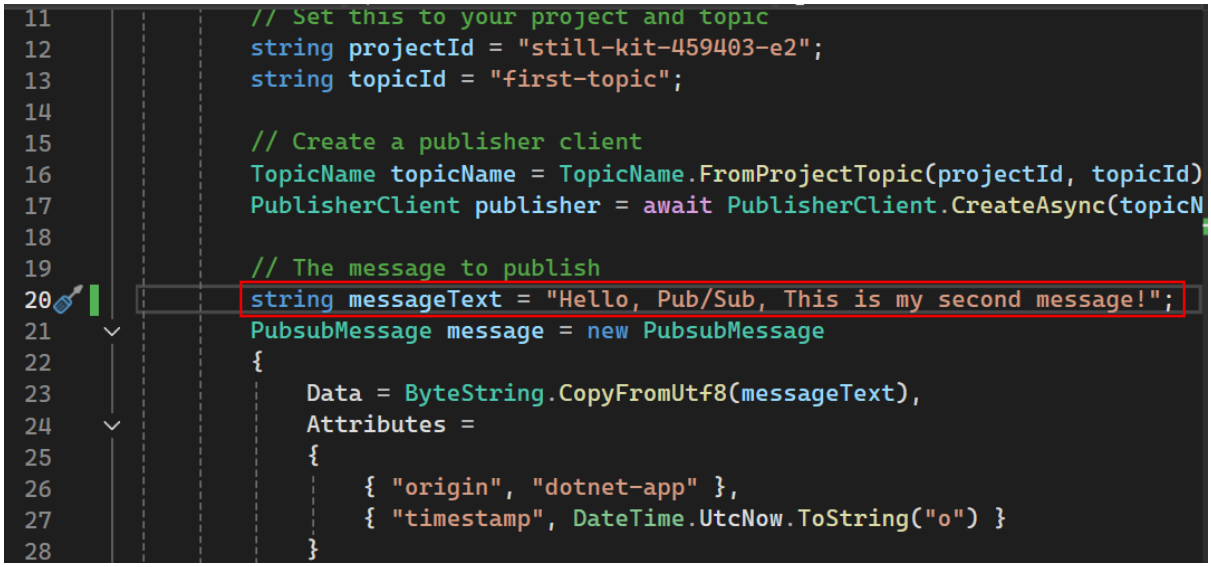


9. Now you need to open the GCP console and navigate to Pub/Sub.

10. Go to messages and from step 2 pull your message, here you can see your message on Pub/Sub. Now you can click on ACK to acknowledge your message.



11. Come back to the application and open the code. On line 20, you can change your message as you can see below.
12. Then just run the program and again repeat the steps to view your message.



```
11          // Set this to your project and topic
12          string projectId = "still-kit-459403-e2";
13          string topicId = "first-topic";
14
15          // Create a publisher client
16          TopicName topicName = TopicName.FromProjectTopic(projectId, topicId)
17          PublisherClient publisher = await PublisherClient.CreateAsync(topicN
18
19          // The message to publish
20          string messageText = "Hello, Pub/Sub, This is my second message!";
21          PubsubMessage message = new PubsubMessage
22          {
23              Data = ByteString.CopyFromUtf8(messageText),
24              Attributes =
25              {
26                  { "origin", "dotnet-app" },
27                  { "timestamp", DateTime.UtcNow.ToString("o") }
28              }
```

13. Here you can see your message, just click on ACK to acknowledge your message.