



Google Cloud Software Development Kit (SDK) for .NET

Author: Luis Coco Enríquez

Personal experience

Senior .Net Software Engineer

Microservices, SQL Server SSMS, SSRS, Oracle SQL Developer, ASP.Net WebForms, ASP.Net MVC5, ASP.Net Core 3.1, .Net 6



<https://www.linkedin.com/in/luis-coco-enriquez-44a28a29/>



CERTIFICATION

Microsoft Certified: Azure Solutions Architect Expert

Expires on November 29, 2023 at 12:59 AM (UTC -0-1:00) • Earned on November 28, 2022



CERTIFICATION

Microsoft Certified: Azure Administrator Associate

Expires on September 19, 2023 at 1:59 AM (UTC -0-2:00) • Earned on September 18, 2021



CERTIFICATION

Microsoft Certified: Azure Developer Associate

Expires on November 2, 2023 at 12:59 AM (UTC -0-1:00) • Earned on November 1, 2021



CERTIFICATION

Microsoft Certified: Azure Fundamentals



CERTIFICATION

Microsoft Certified: Azure AI Fundamentals



CERTIFICATION

Microsoft Certified: Azure Data Fundamentals

Agenda

Topic

What is Google Cloud SDK for .NET?

Key features and components of the Google Cloud SDK or .NET

Create a Google Cloud free tier account

Setting up a .NET development environment

GitHub official samples

Agenda

Topic

Google Cloud guided samples

Create a Bucket, a first simple sample

API reference

Samples in C#

What is Google Cloud SDK for .NET?

The Google Cloud Software Development Kit (SDK) for .NET is a **collection of libraries, tools, and APIs** that enable developers to integrate their .NET applications with various Google Cloud services.

By using the SDK, developers can leverage the power and scalability of Google Cloud services while writing their applications in .NET.

This opens up a wide range of possibilities for building robust and scalable cloud-native applications, leveraging Google's infrastructure and services.

<https://cloud.google.com/dotnet>

<https://cloud.google.com/dotnet/docs/getting-started>

<https://cloud.google.com/dotnet/docs/reference>

Key features and components

1. **Google Cloud Client Libraries:** provides idiomatic .NET client libraries for different Google Cloud services. These libraries abstract the low-level details of interacting with Google Cloud APIs.
2. **Cloud Tools for Visual Studio:** to manage Google Cloud resources, deploy applications to the cloud, and debug their applications directly within the IDE.
3. **Authentication and Authorization:** The SDK provides authentication and authorization mechanisms to securely access Google Cloud services.
4. **Deployment and Management:** It offers tools and APIs to package and deploy your applications, manage your cloud resources, and monitor their performance.
5. **Testing and Emulation:** for testing and emulating Google Cloud services locally, allowing you to develop and test your applications without incurring costs or relying on a live cloud environment. This helps streamline the development and testing process.

Create a Google Cloud free tier account

1. Create a **gmail account**. <https://support.google.com/mail/answer/56256?hl=en>
2. Set your **payment info** (credit card data, paypal,... or other).
3. Open an **incognito Chrome window** and login in Google Cloud.
4. **Login in Google Cloud**. <https://console.cloud.google.com/>
5. Activate the **free tier subscription**.

Setting up a .NET development environment

1. Install your IDE: Visual Studio, VS Code or IntelliJ IDEA Community Edition.

<https://visualstudio.microsoft.com/>

<https://code.visualstudio.com/download>

<https://www.jetbrains.com/idea/download/?section=windows>

2. Create a Google Cloud project. Create a Google Cloud project to run your apps. Google Cloud projects form the basis for creating, enabling, and using all Google Cloud services.

3. Authentication. Your .NET app must authenticate itself to use Google Cloud APIs. You use [Application Default Credentials \(ADC\)](#), which let you provide credentials for either local development or in a production environment.

For information about setting up ADC, see [Provide credentials to Application Default Credentials](#). For general information about authentication, see [Authentication at Google](#).

<https://cloud.google.com/dotnet/docs/setup?hl=en>

Set up Application Default Credentials (ADC)

<https://cloud.google.com/docs/authentication/provide-credentials-adc#local-dev>

1. **Install “gcloud CLI”:** <https://cloud.google.com/sdk/docs/install?hl=en>
2. **Create your credential file** (run the command):

```
gcloud auth application-default login
```

This method stores your credentials in a file on your file system. Any user with access to your file system can use those credentials. When you no longer need these credentials, you should revoke them:

```
gcloud auth application-default revoke
```

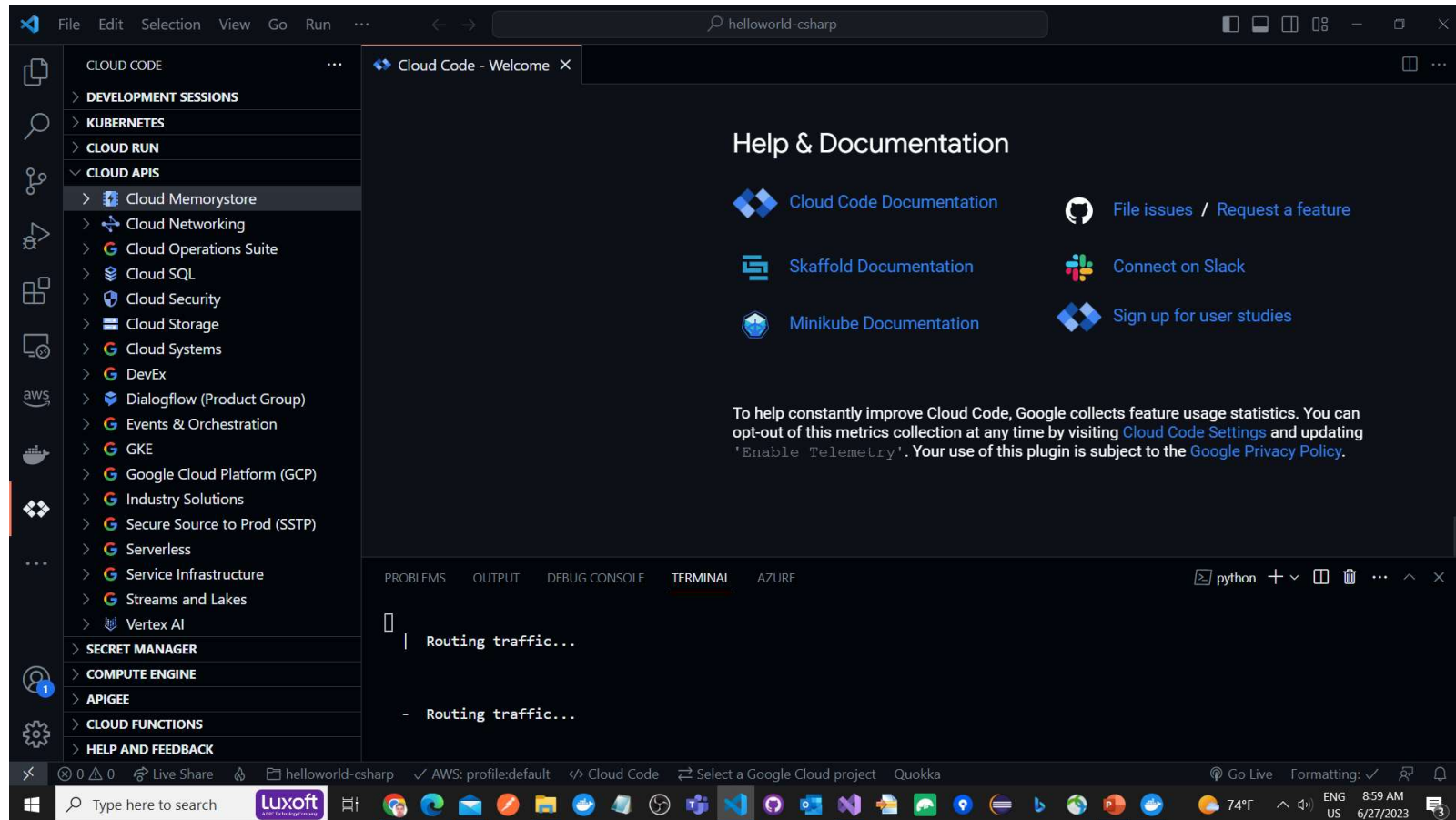
User credentials might not work for some methods and APIs, such as the *Cloud Translation API* or the *Cloud Vision API*, without extra parameters or configuration. See [Troubleshooting your ADC setup](#).

<https://cloud.google.com/dotnet/docs/setup?hl=en>

Google Cloud Code for VSCode

The screenshot shows the Visual Studio Code interface with the Google Cloud Code extension installed. The left sidebar displays the 'EXTENSIONS: MARKETPLACE' view, listing several extensions including Google Cloud Code, Vscode Google Transl..., Google Fonts, Google Translate, google-search, Google Translate, google-java-format, and Google Colab Dark Theme. The main editor area shows the 'Extension: Google Cloud Code' details page for version 1.21.8. The page includes the Google Cloud logo, a description of the extension, and buttons to 'Disable' or 'Uninstall'. A description states: 'Cloud Code is Google Cloud's official IDE extension to develop with your favorite Goog...'. Below this, it says 'This extension is enabled globally.' The page also features tabs for 'DETAILS', 'FEATURE CONTRIBUTIONS', 'CHANGELOG', and 'RUNTIME STATUS'. The 'DETAILS' tab is active, showing a description of the extension's capabilities and a preview of the extension's interface. The right sidebar shows 'Categories' (Snippets, Linters, Debuggers, Other) and 'Extension Resources' (Marketplace, Repository, License, Google Cloud). The bottom status bar shows the current file is 'Cloud Code - Welcome' and the extension is 'Cloud Code: Installing Cloud SDK...'. The system tray at the bottom indicates the date and time as 6/27/2023, 8:17 AM.

Google Cloud Code for VSCode



Google Cloud Code for Visual Studio 2022

The screenshot shows a web browser window displaying the Google Cloud Code for Visual Studio 2022 Codelab. The browser's address bar shows the URL codelabs.developers.google.com/codelabs/cloud-visual-studio#0. The page title is "Install and use Cloud Tools for Visual Studio". On the left, there is a sidebar with a list of steps: 1. Overview, 2. Setup and Requirements, 3. Installation, 4. Authentication, 5. Google Cloud Explorer, 6. Create and Manage Google Compute Engine instances, 7. Create and Deploy ASP.NET apps, 8. Browse Google Cloud Storage, and 9. Congratulations!. The main content area is titled "Install and use Cloud Tools for Visual Studio". It includes a section "Acerca de este codelab" with the text "Última actualización: ene 7, 2021" and "Escrito por Mete Atamel". Below this is a section "1. Overview" with the text: "Visual Studio is the go-to IDE for Windows and .NET developers. Cloud Tools for Visual Studio is a plugin to build and test Windows and .NET applications locally and deploy them to Google Cloud Platform directly from Visual Studio. Google Cloud Explorer lets you easily browse Google Cloud projects and resources across all of your accounts in Visual Studio. In this codelab, you will learn how to install and use Cloud Tools for Visual Studio and how to interact with Google Cloud resources from Visual Studio directly." At the bottom right of the main content area, there is a blue button labeled "Siguiente". The Windows taskbar is visible at the bottom of the browser window, showing various application icons and the system clock.

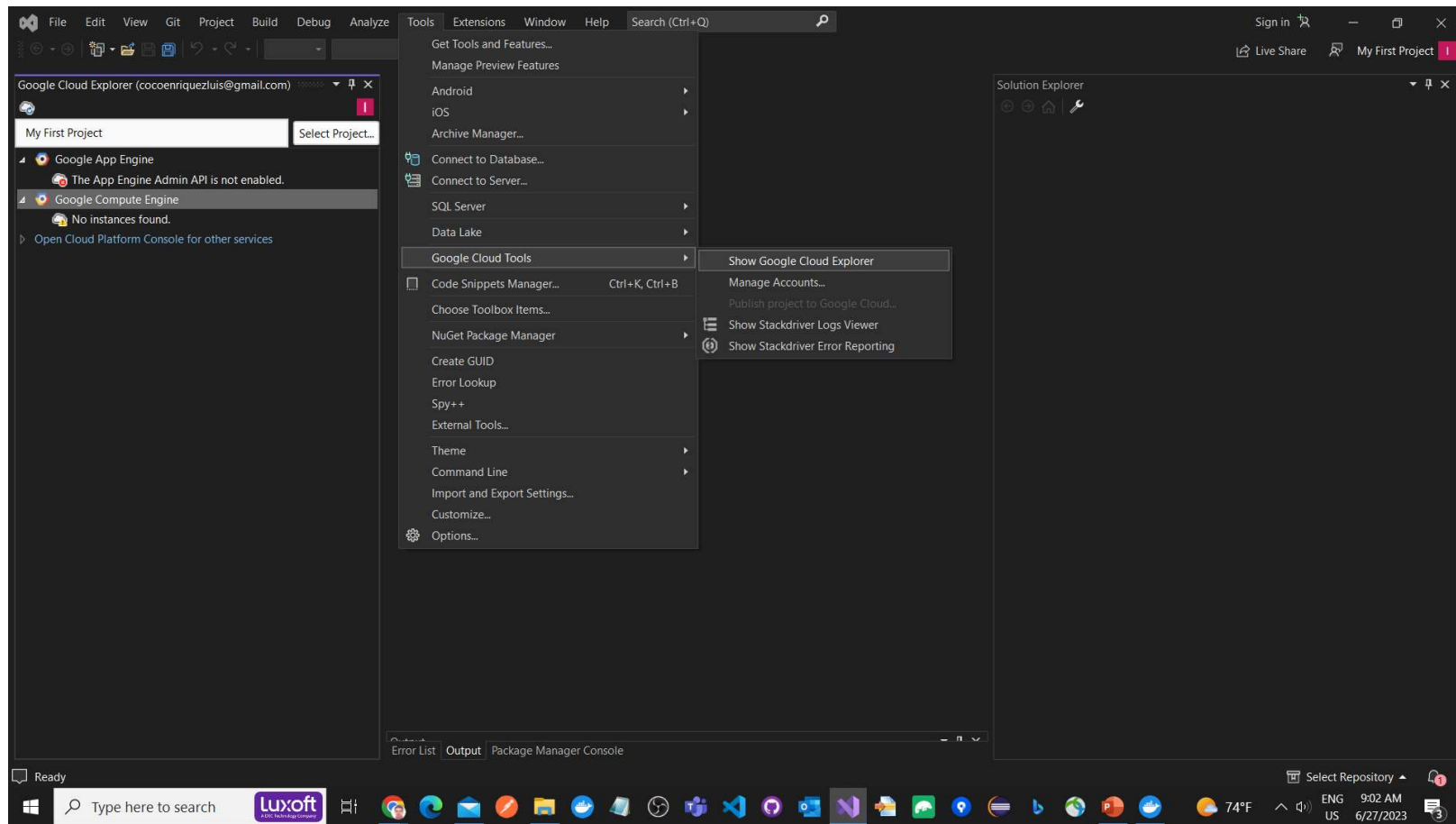
<https://codelabs.developers.google.com/codelabs/cloud-visual-studio#0>

Google Cloud Code for Visual Studio 2022

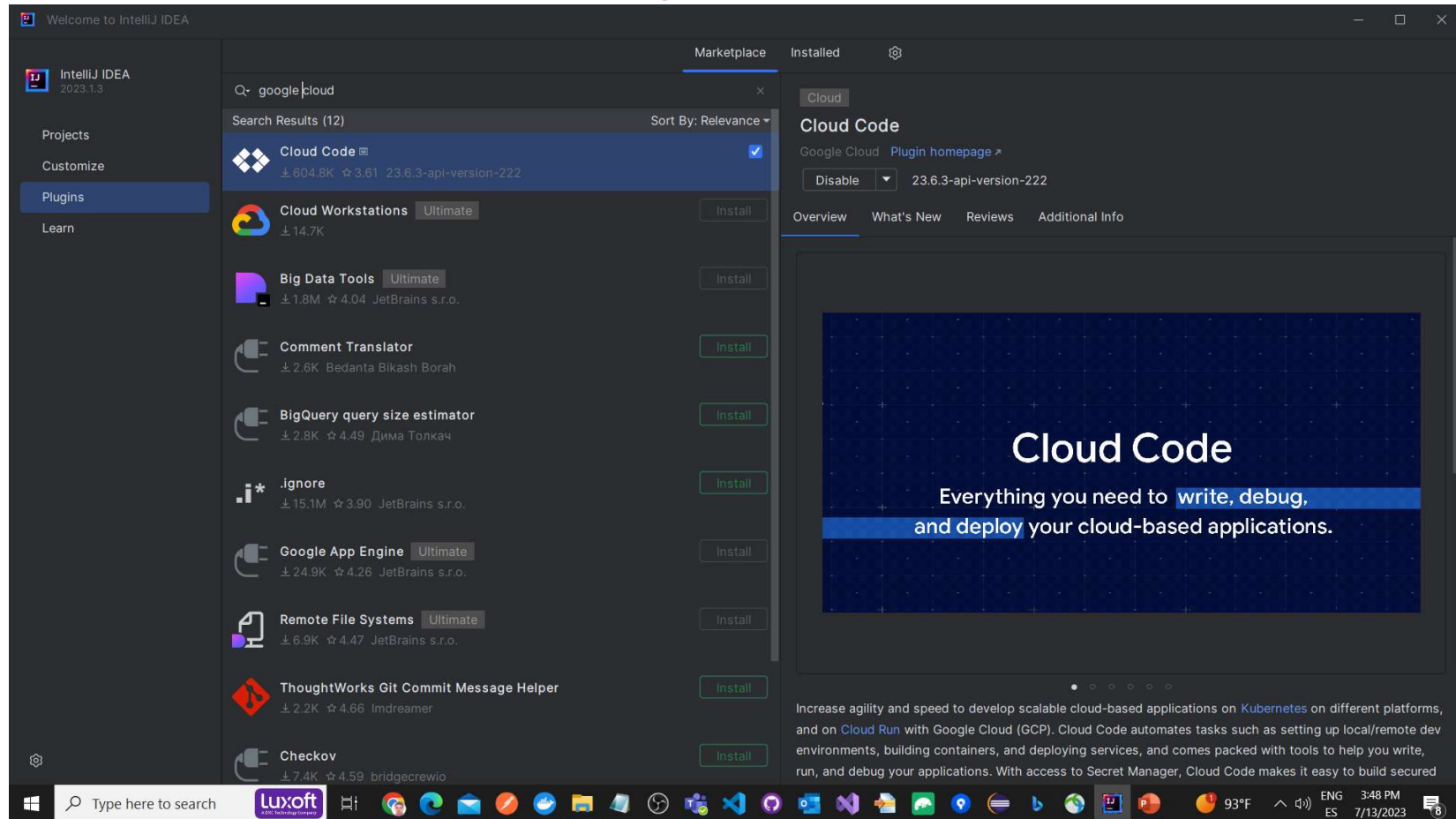
The screenshot shows the Visual Studio Marketplace page for the 'Google Cloud Tools for Visual Studio' extension. The browser's address bar shows the URL: marketplace.visualstudio.com/items?itemName=GoogleCloudTools.GoogleCloudPlatformExtensionforVisualStudio. The page header includes the Visual Studio logo and 'Marketplace' text. Below the header, the extension's name 'Google Cloud Tools for Visual Studio' is displayed, along with the Google Cloud logo, a link to 'google.com', and statistics: '63,437 installs', a star rating of 4.5 (17 reviews), and 'Free'. A green 'Download' button is visible. The 'Overview' tab is selected, showing a description of the extension's capabilities, such as deploying applications to Google Cloud Platform, managing Stackdriver logs, and creating project templates. On the right side, there are sections for 'Categories' (Tools, Build, Web, Services), 'Tags' (ASP.NET, Cloud, compute, Deploy, gce, gcp, gcs, Google, Storage, tool), 'Works with' (Visual Studio 2017, 2019), and 'Resources' (License, Copy ID). The Windows taskbar at the bottom shows the Luxoft logo and various application icons.

<https://marketplace.visualstudio.com/items?itemName=GoogleCloudTools.GoogleCloudPlatformExtensionforVisualStudio>

Google Cloud Code for Visual Studio 2022



IntelliJ IDEA Community Edition



IntelliJ IDEA Community Edition

The screenshot displays the IntelliJ IDEA Marketplace interface. On the left, a sidebar shows navigation options: Projects, Customize, Plugins (selected), and Learn. The main area is titled 'Marketplace' and shows search results for 'google cloud'. The results are sorted by Relevance and include the following plugins:

- Cloud Code** (Selected): 604.8K downloads, 3.61 rating, version 23.6.3-api-version-222. It has an 'Install' button.
- Cloud Workstations** (Ultimate): 14.7K downloads. It has an 'Install' button.
- Big Data Tools** (Ultimate): 1.8M downloads, 4.04 rating, by JetBrains s.r.o. It has an 'Install' button.
- Comment Translator**: 2.6K downloads, by Bedanta Bilash Borah. It has an 'Install' button.
- BigQuery query size estimator**: 2.8K downloads, 4.49 rating, by Дима Толкач. It has an 'Install' button.
- .ignore**: 15.1M downloads, 3.90 rating, by JetBrains s.r.o. It has an 'Install' button.
- Google App Engine** (Ultimate): 24.9K downloads, 4.26 rating, by JetBrains s.r.o. It has an 'Install' button.
- Remote File Systems** (Ultimate): 6.9K downloads, 4.47 rating, by JetBrains s.r.o. It has an 'Install' button.
- ThoughtWorks Git Commit Message Helper**: 2.2K downloads, 4.66 rating, by Imdreamer. It has an 'Install' button.
- Checkov**: 7.4K downloads, 4.59 rating, by bridgecrewio. It has an 'Install' button.

On the right, the details for the 'Cloud Code' plugin are shown. It includes a 'Disable' button, the version '23.6.3-api-version-222', and tabs for Overview, What's New, Reviews, and Additional Info. The 'Key Features' section lists the following:

- Ready-to-go dev environment** - comes with pre-configured tools you need to create, build, test and debug cloud-based applications.
- Real debugging in local or remote clusters** - no more debugging via logs or having to manually set up port forwarding to debug!
- Authentication** - securely connect to Google Cloud using the gcloud command-line tool to access resources.
- Kubernetes** - deploy to clusters, browse Kubernetes resources with one-click access to monitor progress, debug, stream logs, and connect to pod terminals, among a variety of other actions.
- Serverless with Cloud Run** - browse Cloud Run services, deploy to Cloud Run, and debug using a local emulator.
- Compute Engine** - view the Compute Engine VMs running your applications, establish connections to your VMs using SSH, view application logs, and upload files to a VM.
- App Engine** - create a new project to run and debug locally for applications in the standard environment, and to deploy projects in either the standard or flexible environment.
- YAML editing support** - use snippets, linting, smart completion, and automatic YAML validation to easily configure resources.
- Cloud APIs** - access most Google Cloud services with Cloud APIs for Go, Java, Python and Node.js.
- Store application secrets securely with Secret Manager** - create and securely manage secrets by storing sensitive information with Secret Manager instead of in the codebase.
- Google Cloud SQL** - view Cloud SQL instances, connect to a Cloud SQL database locally, and create new IDE data sources.
- Custom Samples** - set up and share repositories of application samples, custom to your organization, to get your team started quickly.

GitHub official samples

GitHub repository page for **GoogleCloudPlatform/dotnet-docs-samples** (Public).

Navigation: <> Code | Issues 30 | Pull requests 13 | Actions | Projects | Wiki | Security | Insights

Repository details: 8 branches, 0 tags, 3,655 commits, 686 stars, 1.2k forks.

Files and folders (commit history):

| File/Folder | Commit Message | Time |
|----------------------|---|--------------|
| .github | fix: Renovate config JSON missing a comma. | 4 days ago |
| .kokoro-windows | chore: Remove blanks | 3 months ago |
| .kokoro | ci: We don't need to order the tests files based on Git timestamp. | last month |
| appengine/flexible | chore(deps): update dependency microsoft.net.test.sdk to v17.6.0 | 2 days ago |
| applications | chore(deps): update dependency microsoft.net.test.sdk to v17.6.0 | 2 days ago |
| asset/api | chore(deps): update dependency microsoft.net.test.sdk to v17.6.0 | 2 days ago |
| auth | chore(deps): update dependency microsoft.net.test.sdk to v17.6.0 | 2 days ago |
| bigquery-storage/api | chore(deps): update dependency microsoft.net.test.sdk to v17.6.0 | 2 days ago |
| bigquery | chore(deps): update dependency microsoft.net.test.sdk to v17.6.0 | 2 days ago |
| bigtable/api | tests: Skip Bigtable samples tests as dependencies are now too old. | last year |
| cdn/signUrl | chore(deps): update dependency microsoft.net.test.sdk to v17.6.0 | 2 days ago |

About: .NET code samples used on <https://cloud.google.com>.
cloud.google.com/dotnet/samples
Readme, Apache-2.0 license, Code of conduct, Security policy, 686 stars, 143 watching, 1.2k forks, Report repository.

Releases: No releases published.

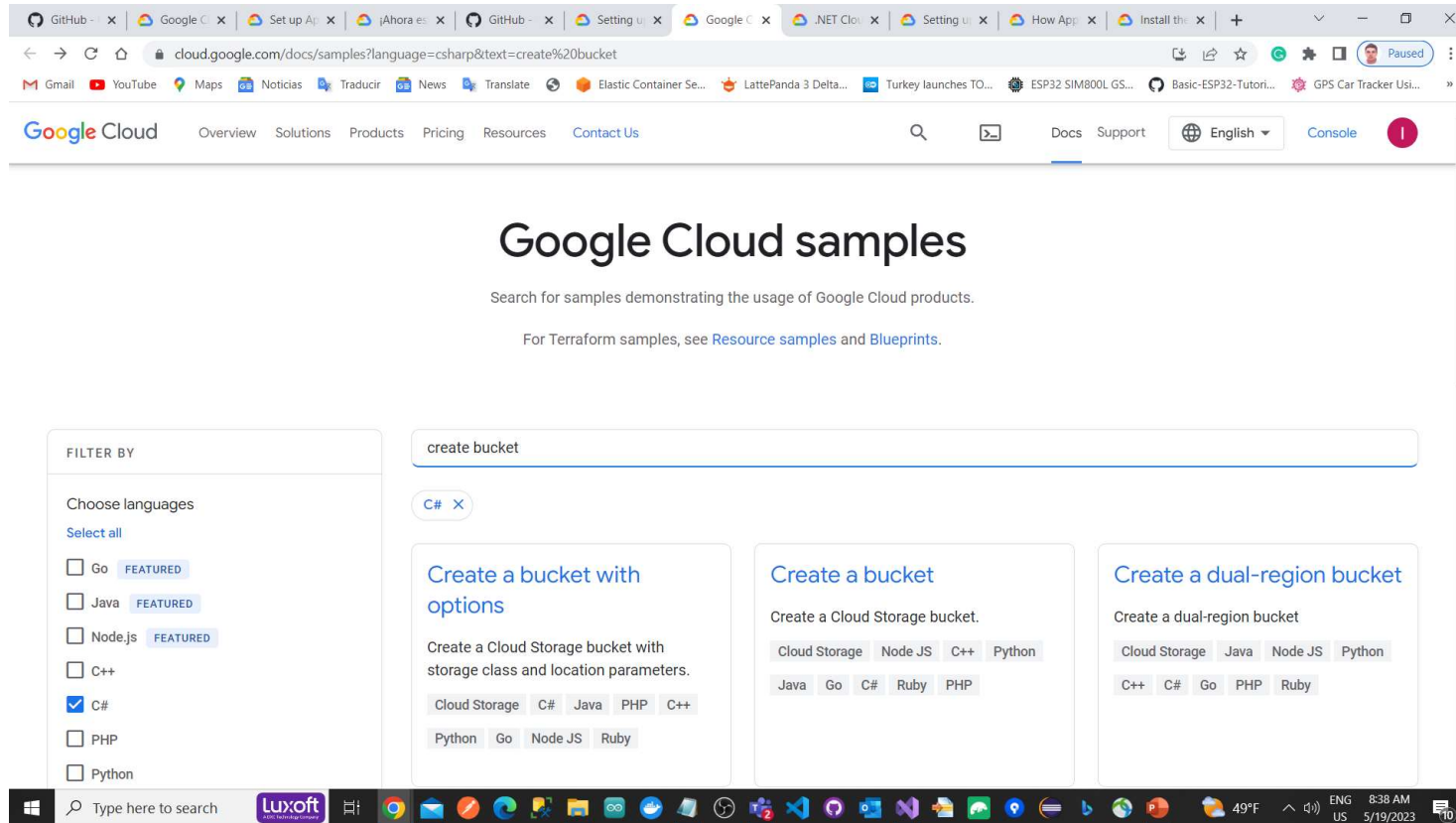
<https://github.com/GoogleCloudPlatform/dotnet-docs-samples>

Google Cloud guided samples

The screenshot shows a web browser window with multiple tabs open. The active tab is 'cloud.google.com/docs/samples?language=csharp'. The page title is 'Google Cloud samples'. Below the title, it says 'Search for samples demonstrating the usage of Google Cloud products.' and 'For Terraform samples, see [Resource samples](#) and [Blueprints](#).' On the left, there is a 'FILTER BY' section with 'Choose languages' and a list of languages: Go (FEATURED), Java (FEATURED), Node.js (FEATURED), C++ (checked), C# (checked), PHP, and Python. The main content area shows three sample cards: 'Update an input endpoint', 'List channels', and 'Delete a channel event'. Each card has a description and a list of supported languages: Live Stream API, Go, Python, Java, PHP, Node JS, C#, and Ruby. The bottom of the screen shows a Windows taskbar with various application icons and a system tray with the date and time (5/19/2023, 8:37 AM).

<https://cloud.google.com/docs/samples?language=csharp>

Create a Bucket, a first simple sample



<https://cloud.google.com/docs/samples?language=csharp&text=create%20bucket>

Create a Bucket, a first simple sample

The screenshot displays the Google Cloud documentation page for creating a bucket. The left sidebar lists various code samples, with 'Create a bucket' selected. The main content area provides instructions on creating a Cloud Storage bucket and includes a link to further documentation. The 'Code sample' section offers a C# example for authenticating to Cloud Storage using Application Default Credentials. The right sidebar provides a 'On this page' summary with links to 'Explore further', 'Code sample', and 'What's next'.

https://cloud.google.com/storage/docs/samples/storage-create-bucket?hl=en#storage_create_bucket-csharp

Create a Bucket, a first simple sample

The screenshot shows a web browser window displaying the Google Cloud Storage C# API reference documentation. The browser's address bar shows the URL: `cloud.google.com/storage/docs/samples/storage-create-bucket?hl=en#storage_create_bucket-csharp`. The page is titled "Create a bucket" and is part of the "Samples" section under "Cloud Storage".

The left sidebar contains a list of code samples for all products, with "Create a bucket" selected. The main content area shows the C# code for creating a bucket. The code is as follows:

```
using Google.Apis.Storage.v1.Data;
using Google.Cloud.Storage.V1;
using System;

public class CreateBucketSample
{
    public Bucket CreateBucket(
        string projectId = "your-project-id",
        string bucketName = "your-unique-bucket-name")
    {
        var storage = StorageClient.Create();
        var bucket = storage.CreateBucket(projectId, bucketName);
        Console.WriteLine($"Created {bucketName}.");
        return bucket;
    }
}
```

The right sidebar contains a "On this page" section with links to "Explore further", "Code sample", and "What's next".

API reference

The screenshot shows a web browser window displaying the Google Cloud .NET Cloud Client Libraries reference page. The browser's address bar shows the URL cloud.google.com/dotnet/docs/reference. The page has a navigation bar with links for Overview, Solutions, Products, Pricing, and Resources. Below this, there's a sub-navigation bar for .NET, Overview, Guides, Reference, and Samples. The main content area is titled ".NET Cloud Client Libraries" and includes a brief description of the libraries and a list of available libraries. A sidebar on the left lists various Google Cloud APIs and their corresponding .NET client libraries.

Google Cloud Overview Solutions Products Pricing Resources

.NET Overview Guides Reference Samples

Filter

Overview

Library reference docs

- Client library help
- Google.Api.CommonProtos
- Google.Api.Gax
- Google.Apis
- Google.Apis.Script.Type
- Google.Cloud.AIPlatform.V1
- Google.Cloud.AccessApproval.V1
- Google.Cloud.AdvisoryNotifications.V1
- Google.Cloud.AlloyDb.V1
- Google.Cloud.AlloyDb.V1Alpha
- Google.Cloud.AlloyDb.V1Beta
- Google.Cloud.ApiGateway.V1
- Google.Cloud.ApiKeys.V2
- Google.Cloud.ApigeeConnect.V1
- Google.Cloud.ApigeeRegistry.V1
- Google.Cloud.AppEngine.Logging.V1

.NET > Documentation > Reference

Was this helpful?

.NET Cloud Client Libraries

The Cloud Client Libraries are the recommended way to access Google Cloud APIs programmatically. The Cloud Client Libraries support accessing Google Cloud services in a way that significantly reduces the boilerplate code you have to write. The libraries provide high-level API abstractions so they're easier to understand. They embrace idioms of the language, work well with the standard library, and integrate better with your codebase.

For more information about client libraries for Cloud APIs, see [Cloud Client Libraries explained](#).

Click an API to find out how to use it in .NET:

Search libraries and products

| Libraries | |
|------------------------|--|
| Access Approval | Google.Cloud.AccessApproval.V1 |
| Access Context Manager | Google.Identity.AccessContextManager.Type Google.Identity.AccessContextManager.V1 |
| Advisory Notifications | Google.Cloud.AdvisoryNotifications.V1 |

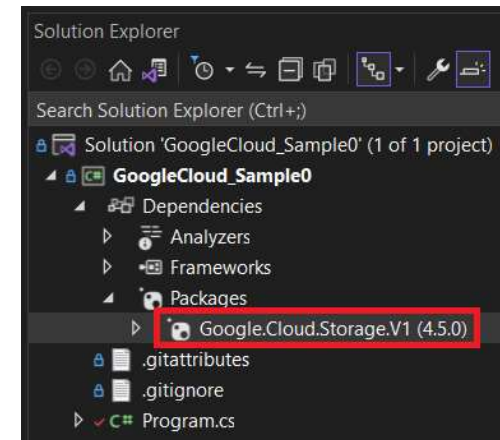
<https://cloud.google.com/dotnet/docs/reference>

Sample 1 Buckets (create/list) Files upload/download

```
using Google.Apis.Storage.v1.Data;
using Google.Cloud.Storage.V1;

string projectId = "focus-cache-387205";
string bucketName = "luiscocoenriquezsegundo";
string localPath = "C://NetChapterArticles.txt";
string objectName = "NetChapterArticles.txt";

//-----
//Create Bucket
//-----
var storage = StorageClient.Create();
var bucket = storage.CreateBucket(projectId, bucketName);
Console.WriteLine($"Created {bucketName}.");
//-----
//List Buckets
//-----
//var storage = StorageClient.Create();
var bucketsluis = storage.ListBuckets(projectId);
Console.WriteLine("Buckets:");
foreach (var bucketluis in bucketsluis)
{
    Console.WriteLine(bucketluis.Name);
}
//-----
//File Upload
//-----
using var fileStream = File.OpenRead(localPath);
storage.UploadObject(bucketName, objectName, null, fileStream);
Console.WriteLine($"Uploaded {objectName}.");
```



```
//-----
//File Download
//-----
string downloadlocalpath = "C:\\New folder\\DownloadedFile.txt";
using var outputFile = File.OpenWrite(downloadlocalpath);
storage.DownloadObject(bucketName, objectName, outputFile);
Console.WriteLine($"Downloaded {objectName} to {localPath}.");
//-----
//List Files
//-----
var storageObjects = storage.ListObjects(bucketName);
Console.WriteLine($"Files in bucket {bucketName}:");
foreach (var storageObject in storageObjects)
{
    Console.WriteLine(storageObject.Name);
}
```

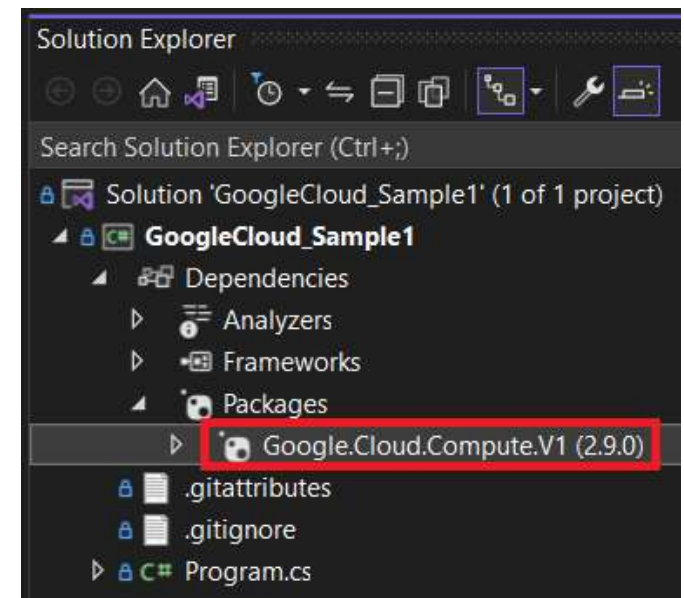
Sample 2 Create a Virtual Machine

```
using Google.Cloud.Compute.V1;

string projectId = "focus-cache-387205";
string zone = "europe-southwest1-a";
string machineName = "luis-test-machine";
string machineType = "e2-micro";
string diskImage = "projects/debian-cloud/global/images/family/debian-10";
long diskSizeGb = 10;
string networkName = "default";

Instance instance = new Instance
{
    Name = machineName,
    MachineType = $"zones/{zone}/machineTypes/{machineType}",
    Disks =
    {
        new AttachedDisk
        {
            AutoDelete = true,
            Boot = true,
            Type = ComputeEnumConstants.AttachedDisk.Type.Persistent,
            InitializeParams = new AttachedDiskInitializeParams
            {
                SourceImage = diskImage,
                DiskSizeGb = diskSizeGb
            }
        }
    },
    NetworkInterfaces = { new NetworkInterface { Name = networkName } }
};
```

```
InstancesClient client = await InstancesClient.CreateAsync();
var instanceCreation = await client.InsertAsync(projectId, zone, instance);
await instanceCreation.PollUntilCompletedAsync();
```



Sample 3 Deleting a Virtual Machine

```
using Google.Api;
using Google.Cloud.Compute.V1;

string projectId = "focus-cache-387205";
string zone = "europe-southwest1-a";
string machineName = "luis-test-machine";

InstancesClient client = await InstancesClient.CreateAsync();

// Stop the VM instance before deleting it.
var stopRequest = new StopInstanceRequest
{
    Project = projectId,
    Zone = zone,
    Instance = machineName
};

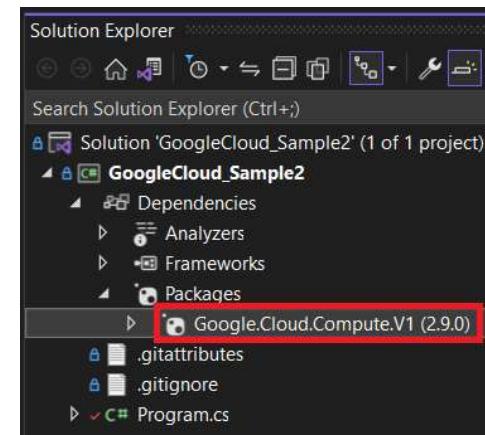
await client.StopAsync(stopRequest);
```

```
// Start the VM instance before deleting it.
//var startRequest = new StartInstanceRequest
//{
//    Project = projectId,
//    Zone = zone,
//    Instance = machineName
//};

//await client.StartAsync(startRequest);

// Make the request to delete a VM instance.
var instanceDeletion = await client.DeleteAsync(projectId, zone, machineName);

//Wait for the operation to complete using client-side polling.
await instanceDeletion.PollUntilCompletedAsync();
```



Sample 4 List Virtual Machines and Virtual Machine Types

Listing Virtual Machines

```
using Google.Cloud.Compute.V1;

string projectId = "focus-cache-387205";

InstancesClient client = await InstancesClient.CreateAsync();
IList<Instance> allInstances = new List<Instance>();

// Make the request to list all VM instances in a project.
await foreach (var instancesByZone in client.AggregatedListAsync(projectId))
{
    Console.WriteLine($"Instances for zone: {instancesByZone.Key}");
    foreach (var instance in instancesByZone.Value.Instances)
    {
        Console.WriteLine($"-- Name: {instance.Name}");
        allInstances.Add(instance);
    }
}
```

Listing Virtual Machines Types

```
using Google.Cloud.Compute.V1;

MachineTypesClient machineTypesClient = MachineTypesClient.Create();
ImagesClient imagesClient = ImagesClient.Create();

// List machine types
var machineTypesList = machineTypesClient.List(new ListMachineTypesRequest
{
    Project = "focus-cache-387205",
    Zone = "europe-southwest1-a"
});

Console.WriteLine("Machine Types:");
foreach (MachineType machineType in machineTypesList)
{
    Console.WriteLine($"- {machineType.Name}");
}

Console.WriteLine();
```

Sample 5 Datastore

```
using Google.Cloud.Datastore.V1;

string projectId = "focus-cache-387205";

DatastoreDb db = DatastoreDb.Create(projectId);

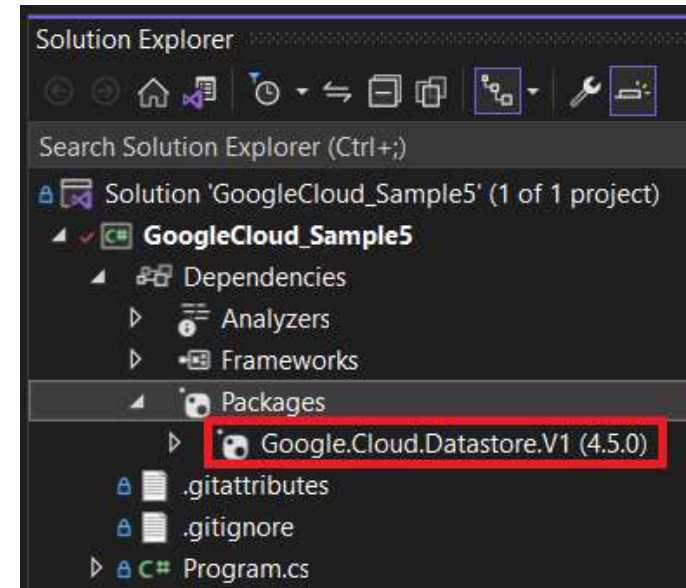
string kind = "Task";
string name = "sampletask1";

KeyFactory keyFactory = db.CreateKeyFactory(kind);
Key key = keyFactory.CreateKey(name);

var task = new Entity
{
    Key = key,
    ["description"] = "Buy milk"
};

using (DatastoreTransaction transaction = db.BeginTransaction())
{
    // Saves the task
    transaction.Upsert(task);
    transaction.Commit();

    Console.WriteLine($"Saved {task.Key.Path[0].Name}: {(string)task["description"]}");
}
```



Sample 6 Firestore

```
using Google.Cloud.Firestore;

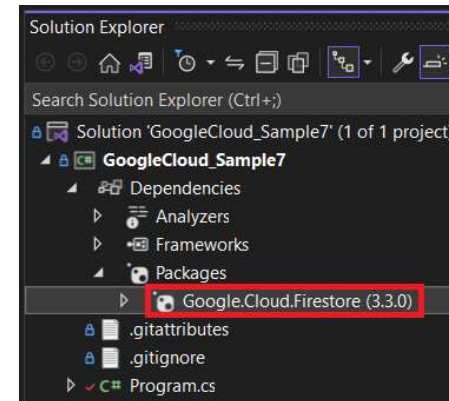
string project = "focus-cache-387205";

AddData1(project).Wait();
RetrieveAllDocuments(project).Wait();

void InitializeProjectId(string project)
{
    FirestoreDb db = FirestoreDb.Create(project);
    Console.WriteLine("Created Cloud Firestore client with project ID: {0}",
project);
}

async Task AddData1(string project)
{
    FirestoreDb db = FirestoreDb.Create(project);
    DocumentReference docRef = db.Collection("users").Document("alovelace");
    Dictionary<string, object> user = new Dictionary<string, object>
    {
        { "First", "Ada" },
        { "Last", "Lovelace" },
        { "Born", 1815 }
    };
    await docRef.SetAsync(user);
    Console.WriteLine("Added data to the avelace document in the users
collection.");
}
```

```
async Task RetrieveAllDocuments(string project)
{
    FirestoreDb db = FirestoreDb.Create(project);
    CollectionReference usersRef = db.Collection("users");
    QuerySnapshot snapshot = await usersRef.GetSnapshotAsync();
    foreach (DocumentSnapshot document in snapshot.Documents)
    {
        Console.WriteLine("User: {0}", document.Id);
        Dictionary<string, object> documentDictionary = document.ToDictionary();
        Console.WriteLine("First: {0}", documentDictionary["First"]);
        if (documentDictionary.ContainsKey("Middle"))
        {
            Console.WriteLine("Middle: {0}", documentDictionary["Middle"]);
        }
        Console.WriteLine("Last: {0}", documentDictionary["Last"]);
        Console.WriteLine("Born: {0}", documentDictionary["Born"]);
        Console.WriteLine();
    }
}
```



Sample 7 Pub/Sub

```
using Google.Cloud.PubSub.V1;
using Grpc.Core;

string projectId = "focus-cache-387205";
string topicId = "luis-topic-1";
string subscriptionId = "subscription-first";

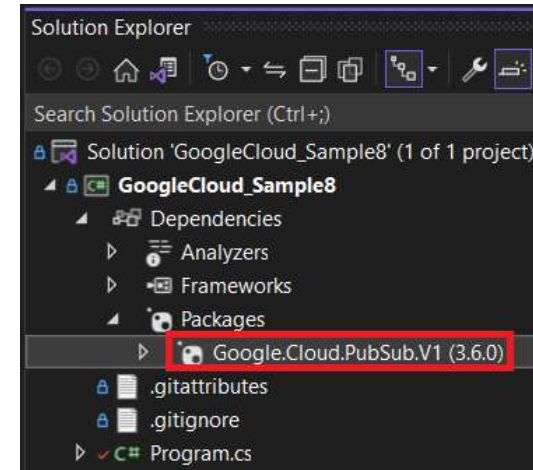
//Create a topic
PublisherServiceApiClient publisher = PublisherServiceApiClient.Create();
var topicName = TopicName.FromProjectTopic(projectId, topicId);
Topic topic = null;

try
{
    topic = publisher.CreateTopic(topicName);
    Console.WriteLine($"Topic {topic.Name} created.");
}
catch (RpcException e) when (e.Status.StatusCode == StatusCode.AlreadyExists)
{
    Console.WriteLine($"Topic {topicName} already exists.");
}
```

```
//Create a Subscription for the topic
SubscriberServiceApiClient subscriber = SubscriberServiceApiClient.Create();

SubscriptionName subscriptionName =
SubscriptionName.FromProjectSubscription(projectId, subscriptionId);
Subscription subscription = null;

try
{
    subscription = subscriber.CreateSubscription(subscriptionName, topicName,
        pushConfig: null, ackDeadlineSeconds: 60);
}
catch (RpcException e) when (e.Status.StatusCode == StatusCode.AlreadyExists)
{
    // Already exists. That's fine.
}
```



Sample 7 Pub/Sub

Pub/Sub

```
//Publish messages to the above created topic
```

```
PublisherClient publisher1 = await PublisherClient.CreateAsync(topicName);

int publishedMessageCount = 0;

List<string> messageTexts = new List<string>();
messageTexts.Add("First message");
messageTexts.Add("Second message");
messageTexts.Add("Third message");
messageTexts.Add("Fourth message");
messageTexts.Add("Fifth message");

var publishTasks = messageTexts.Select(async text =>
{
    try
    {
        string message = await publisher1.PublishAsync(text);
        Console.WriteLine($"Published message {message}");
        Interlocked.Increment(ref publishedMessageCount);
    }
    catch (Exception exception)
    {
        Console.WriteLine($"An error occurred when publishing message {text}: {exception.Message}");
    }
});
await Task.WhenAll(publishTasks);
```

Sample 7 Pub/Sub

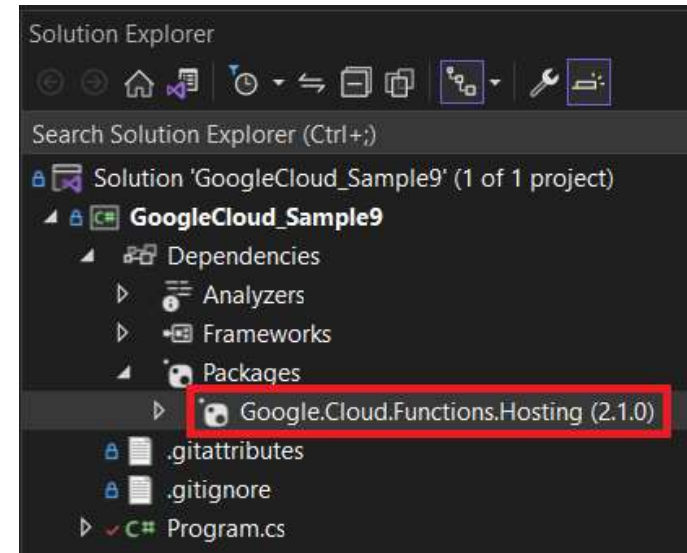
Pub/Sub

```
int messages_Count_received = PullMessagesSync(projectId, subscriptionId, true);

int PullMessagesSync(string projectId, string subscriptionId, bool acknowledge)
{
    SubscriptionName subscriptionName = SubscriptionName.FromProjectSubscription(projectId, subscriptionId);
    SubscriberServiceApiClient subscriberClient = SubscriberServiceApiClient.Create();
    int messageCount = 0;
    try
    {
        // Pull messages from server,
        PullResponse response = subscriberClient.Pull(subscriptionName, maxMessages: 20);
        foreach (ReceivedMessage msg in response.ReceivedMessages)
        {
            string text = System.Text.Encoding.UTF8.GetString(msg.Message.Data.ToArray());
            Console.WriteLine($"Message {msg.Message.MessageId}: {text}");
            Interlocked.Increment(ref messageCount);
        }
        if (acknowledge && messageCount > 0)
        {
            subscriberClient.Acknowledge(subscriptionName, response.ReceivedMessages.Select(msg => msg.AckId));
        }
    }
    catch (RpcException ex) when (ex.Status.StatusCode == StatusCode.Unavailable)
    {
        // UNAVAILABLE due to too many concurrent pull requests pending for the given subscription.
    }
    return messageCount;
}
```

Sample 8 Functions

```
using Google.Cloud.Functions.Framework;  
using Microsoft.AspNetCore.Http;  
  
namespace HelloWorld;  
  
public class Function : IHttpFunction  
{  
    public async Task HandleAsync(HttpContext context)  
    {  
        await context.Response.WriteAsync("Hello World!");  
    }  
}
```



<https://cloud.google.com/functions/docs/concepts/overview>

Sample 8 Functions

```
using CloudNative.CloudEvents;
using Google.Cloud.Functions.Framework;
using Google.Events.Protobuf.Cloud.Storage.V1;
using Microsoft.Extensions.Logging;
using System.Threading;
using System.Threading.Tasks;

namespace HelloGcs;

/// <summary>
/// Example Cloud Storage-triggered function.
/// This function can process any event from Cloud Storage.
/// </summary>
public class Function : ICloudEventFunction<StorageObjectData>
{
    private readonly ILogger _logger;

    public Function(ILogger<Function> logger) =>
        _logger = logger;

    public Task HandleAsync(CloudEvent cloudEvent, StorageObjectData data,
        CancellationToken cancellationToken)
    {
        _logger.LogInformation("Event: {event}", cloudEvent.Id);
        _logger.LogInformation("Event Type: {type}", cloudEvent.Type);
        _logger.LogInformation("Bucket: {bucket}", data.Bucket);
        _logger.LogInformation("File: {file}", data.Name);
        _logger.LogInformation("Metageneration: {metageneration}", data.Metageneration);
        _logger.LogInformation("Created: {created:s}",
            data.TimeCreated?.ToDateTimeOffset());
        _logger.LogInformation("Updated: {updated:s}", data.Updated?.ToDateTimeOffset());
        return Task.CompletedTask;
    }
}
```

