



Cloud Client Libraries,
the Cloud SDK, and
Firebase SDKs

Course name: Developing Applications with Google Cloud Platform

Module name: Cloud Client Libraries, the Cloud SDK, and Firebase SDKs

Featured products: Cloud Client Libraries, Cloud SDK, Firebase SDK

Qwiklabs: Setting up a development environment

What are the Cloud Client Libraries?

Client Libraries:

- Are the latest and recommended way to make requests to the server.
- Provide idiomatic code in each language.
- Receive performance benefits from gRPC APIs.

Cloud Client Libraries are the latest and recommended approach to making requests to the server. The Cloud Libraries make it easier to access API calls using your favorite programming language.

Cloud Client Libraries handle low-level communication with the server, including authentication with Google, and can be installed using familiar installation packages such as npm and pip. The Client Libraries also provide retry logic for transient network failures. Consult the Client Libraries documentation for more details.

Cloud Client Libraries provide idiomatic code in supported languages, which makes them easier to work with. Some libraries give you performance benefits from gRPC. Google Remote Procedure Calls (gRPC) is an open-source remote procedure call framework that can be run anywhere. gRPC makes it easier to build connected systems because it enables client and server applications to communicate transparently.

Google API Client Libraries should only be used if your programming language of choice isn't supported by the Cloud Client Libraries yet. They provide access to REST APIs only and do not support gRPC.

gRPC APIs: https://cloud.google.com/apis/docs/client-libraries-explained#grpc_apis
Authentication with Google: <https://cloud.google.com/docs/authentication/>

What languages are supported by the Cloud Client Libraries?



.NET



Go



Java



Node.js



PHP



Python



Ruby

Cloud Client Libraries are the latest and recommended client libraries for calling Google Cloud APIs. Supported languages include .NET, Go, Java, Node.js, PHP, Python, and Ruby.

<https://cloud.google.com/apis/docs/cloud-client-libraries>

Get started with the Cloud Client Libraries

GitHub repos:

- Provided for each supported language and individual services
- Contain installation instructions and Client Library code

Reference libraries:

- Link to documentation
- Link to relevant StackOverflow posts
- Provide code examples

Google Cloud Client Library	Installation & Reference
Go	<ul style="list-style-type: none">• GitHub Repo• Library Reference
Java	<ul style="list-style-type: none">• GitHub Repo• Library Reference
Node.js	<ul style="list-style-type: none">• GitHub Repo• Library Reference
Python	<ul style="list-style-type: none">• GitHub Repo• Library Reference
Ruby	<ul style="list-style-type: none">• GitHub Repo• Library Reference
PHP	<ul style="list-style-type: none">• GitHub Repo• Library Reference
C#	<ul style="list-style-type: none">• GitHub Repo• Library Reference

You can pull the repo for the Cloud Client Libraries for each of the supported programming languages. The GitHub Repo page lists the services/APIs supported by each language's Cloud Client library and provides installation instructions. You can also download Cloud Client Libraries for individual Google Cloud services. Reference libraries contain links to documentation and relevant StackOverflow posts and provide code examples. The reference libraries are your one-stop shop for information on a language-specific Cloud Client Library.

For direct links to GitHub Repos and Reference Libraries, see <https://cloud.google.com/apis/docs/cloud-client-libraries>.

Import the Datastore client library using Python

```
from google.cloud import datastore
datastore_client = datastore.Client()
kind = 'Task'
name = 'sampletask1'
task_key = datastore_client.key(kind, name)
task = datastore.Entity(key=task_key)
task['description'] = 'Buy milk'
datastore_client.put(task)
print('Saved {}: {}'.format(task.key.name,
task['description']))
```

Imports the Cloud Client Library

Instantiates a client

The kind and name/ID for the new entity

Creates a Datastore key for the new entity

Prepares the new entity

Saves the entity

Every package uses a Client as a base for interacting with an API. If your application is running on App Engine or Compute Engine, authentication for your application will “just work.” If you don’t explicitly provide credentials, the Client will reuse the credentials from the gcloud tool, assuming it has already been installed and authorized. The example demonstrates importing the Datastore client library, instantiating the client using default credentials, and adding an entity to the Datastore.

Google Application Default Credentials:

<https://developers.google.com/identity/protocols/application-default-credentials>

Authentication with the Python Client Library:

<https://googleapis.dev/python/google-api-core/latest/auth.html>

The Cloud SDK is a set of command-line tools

The Cloud SDK:

- Allows you to access Google Cloud products and services
- Consists of three command-line tools:
 - gcloud, bq, and gsutil
- Allows you to run tools:
 - Interactively
 - In your automated scripts

The Cloud SDK consists of three command-line tools: gcloud, bq, and gsutil. These tools allow you to access Google Cloud products and services. You can run your tools interactively or in your automated scripts.

Cloud SDK Documentation: <https://cloud.google.com/sdk/docs/>

Cloud SDK: gcloud



- A command-line tool
- Allows you to perform common tasks on Google Cloud
- Allows you to create and manage Google Cloud resources

```
gcloud compute instances list
```

NAME	ZONE	MACHINE_TYPE	INTERNAL_IP	EXTERNAL_IP	STATUS
example-instance	asia-east1-b	n1-standard-1	10.240.95.199	107.167.182.44	RUNNING
example-instance2	us-central1-a	n1-standard-1	10.240.173.254	23.251.148.121	RUNNING
test-instance	us-central1-a	n1-standard-1	10.240.118.207	23.251.153.172	RUNNING

The Cloud SDK is a set of command-line tools that helps you manage resources and applications hosted on Google Cloud.

You can perform many common tasks on Google Cloud using the gcloud command-line tool, including creating and managing resources for various services. Alpha and Beta commands provide additional functionality.

This example lists all the Compute Engine VM instances for your project.

<https://cloud.google.com/sdk/docs/overview>

<https://cloud.google.com/sdk/gcloud/>

<https://cloud.google.com/appengine/docs/standard/java/tools/migrating-from-appcfg-to-gcloud>

Cloud SDK: bq



- A command-line tool to work with BigQuery
- Allows you to manage datasets, tables, and other BigQuery entities
- Allows you to run queries

```
bq query "SELECT word, SUM(word_count) as count FROM publicdata:samples.shakespeare WHERE word CONTAINS 'raisin' GROUP BY word"
```

```
Waiting on job_dcda37c0bbbed4c669b04dfd567859b90 ... (0s) Current status: DONE
```

word	count
Praising	4
raising	5
raisins	1
praising	8
dispraising	2
dispraisingly	1

bq is a command-line tool used to work with BigQuery. bq's primary purpose is running queries and it can also be used to manage datasets, tables, and other BigQuery entities.

<https://cloud.google.com/bigquery/bq-command-line-tool>

Cloud SDK: gsutil



- A command-line tool to perform tasks in Cloud Storage
- Allows you to:
 - Create and manage buckets
 - Upload, download, and delete objects
 - Move, copy, and rename objects
 - Manage access to stored data

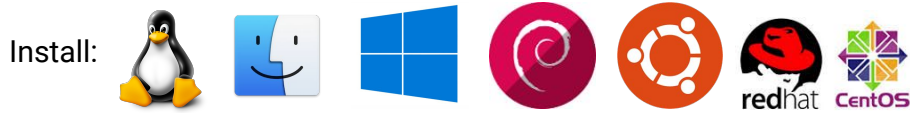
```
gsutil cp Desktop/cloud-storage-logo.png gs://my-awesome-bucket

Copying file://Desktop/cloud-storage-logo.png [Content-Type=image/png]...
Uploading    gs://my-awesome-bucket/cloud-storage-logo.png:      0 B/2.58 KiB
Uploading    gs://my-awesome-bucket/cloud-storage-logo.png:      2.58 KiB/2.58 KiB
```

gsutil is a command-line tool used to perform tasks in Cloud Storage. You can use gsutil to create and manage buckets; upload, download, and delete objects; move, copy, and rename objects; and manage access to stored objects.

<https://cloud.google.com/storage/docs/gsutil>

Installing and configuring the Cloud SDK



Initialize:

```
gcloud init
```

Use:

- Manage components
- Use the gcloud interactive shell (Beta)
- Script gcloud commands

You can download and install the Cloud SDK on Linux, Mac OS X, and Windows. You can install the Cloud SDK using apt-get on Debian and Ubuntu.

Initialize the Cloud SDK by running gcloud init. Once it is initialized, start using it! You can install and manage SDK components and use the gcloud interactive shell (in beta), which provides prompt completion and suggests flags. You can even script gcloud commands to automate your processes.

Downloads: <https://cloud.google.com/sdk/downloads>

Cloud Shell



- A browser-based command-line tool
- Gives you access to a temporary virtual machine instance with:
 - 5 GB of persistent disk storage
 - The Cloud SDK is pre-installed
- Provides built-in authorization to Cloud Console projects and resources
- Has a built-in code editor



Cloud Shell is a free admin machine with browser-based command-line access. It allows you to easily manage your infrastructure and application on Google Cloud. It gives you access to a temporary virtual machine instance with 5 GB of persistent disk storage. When you start Cloud Shell, it provisions an f1-micro Compute Engine virtual machine running a Debian-based Linux OS. Cloud Shell instances are provisioned on a per-user, per-session basis. The instances persist only while your Cloud Shell session is active and terminate after an hour of inactivity. The Cloud SDK comes pre-installed in Cloud Shell. Additionally, Cloud Shell has built-in authorization to your Cloud Console projects and resources. Cloud Shell comes with a built-in code editor based on Orion to browse file directories and view and edit files, with continued access to Cloud Shell. Cloud Shell provides pre-installed language support for Java, Go, Python, Node.js, Ruby, PHP, and .NET.

Integrate with your favorite IDEs to accelerate your development on Google Cloud!

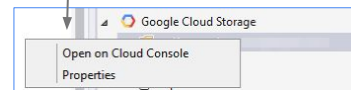
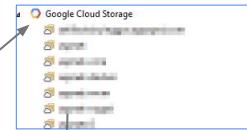
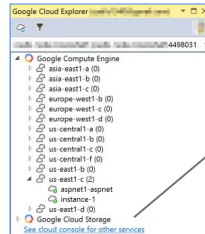
 Android Studio

 Eclipse

 IntelliJ

 PowerShell

 Visual Studio



Google Cloud provides cloud tools for various IDEs to facilitate development on Google Cloud. Cloud tools are supported for Android Studio, Eclipse, IntelliJ, Powershell, and Visual Studio.

For example, Cloud Developer Tools for Visual Studio allows the developer to browse computer engine resources, storage buckets, and Cloud SQL instances from the IDE.

Cloud Developer Tools for Android Studio:

<https://cloud.google.com/tools/android-studio/docs/>

Cloud Developer Tools for Eclipse: <https://cloud.google.com/eclipse/docs/>

Cloud Developer Tools for IntelliJ: <https://cloud.google.com/tools/intellij/docs/>

Cloud Developer Tools for PowerShell (Beta):







<https://cloud.google.com/tools/powershell/docs/>

Cloud Developer Tools for Visual Studio:

<https://cloud.google.com/tools/visual-studio/docs/>

Firebase



- A mobile and web application development platform
- Supported platforms:
 - Android 
 - iOS 
 - Web 
 - C++ 
 - Unity 
 - Node.js 
- Integration with Google Cloud includes:
 - Firebase SDKs for Cloud Storage
 - App Engine standard environment + Firebase
 - Trusted execution added to your Firebase app
 - User authentication
 - Cloud Functions for Firebase
 - Vision API
 - Speech APIs

Firebase is a mobile and web application development platform. Supported platforms include Android, iOS, Web, C++, Unity, and Node.js.

Firebase is integrated with various services in Google Cloud. The Firebase SDKs for Cloud Storage store files directly in Cloud Storage buckets, and you can use the Cloud Storage APIs to access files uploaded via Firebase SDKs for Cloud Storage.

The Firebase SDKs for Cloud Storage use the default bucket for App Engine standard environment, so you can use the built-in App Engine APIs to share data between Firebase and your App Engine app. Additionally, you can retrieve, verify, and store user credentials using Firebase Authentication, the App Engine standard environment, and Datastore.

Cloud Functions is Google Cloud's serverless offering. Cloud Functions for Firebase lets you automatically run backend code in response to events triggered by Firebase features and HTTPS requests, and the code is stored in Google's cloud and run in a managed environment; no management or scaling of your own servers is required!

Other use cases include integrating your Firebase app with Vision and Speech APIs.

<https://firebase.google.com/docs/>

<https://firebase.google.com/docs/storage/gcp-integration>

<https://cloud.google.com/appengine/docs/standard/python/authenticating-users-firebase-appengine>

<https://firebase.google.com/docs/functions/>

<https://cloud.google.com/solutions/mobile/firebase-app-engine-android-studio>

<https://firebase.google.com/docs/storage/gcp-integration>



Google APIs Explorer

1. Navigate to the API explorer
2. Enable an API
3. Invoke an API
4. Review the result of invoking the API

Why the Google APIs Explorer?

- Browse quickly through available APIs and versions.
- See methods available for each API and what parameters they support.
- View inline documentation for methods.
- Execute requests for any method and see responses in real time.
- Make authenticated and authorized API calls.
- Search across all services, methods, and your recent requests.



Setting up a Development Environment

Duration: 30 minutes

Lab objectives

Provision a Compute Engine instance

Connect to the instance using SSH

Install the software on the instance

Verify the software installation

