# Developing Applications with Google Cloud: Foundations

Module 2: Getting started with Google Cloud development

Welcome to Developing Applications with Google Cloud: Foundations, module 2: Getting started with Google Cloud development.

Google Cloud provides many platforms that you can use to build your applications. Your apps can benefit from many powerful services provided by Google Cloud.

## Agenda

| | |
|---|---|
| 01 | Cloud APIs and the Google Cloud SDK |
| 02 | Google Cloud CLI |
| 03 | Cloud Client Libraries |
| 04 | Cloud Shell and Cloud Code |

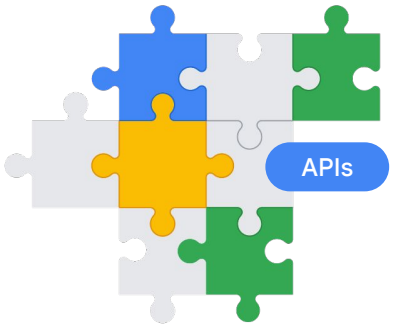
In this module, you learn how to access these services for your apps and scripts. You learn about Cloud APIs and the Google Cloud SDK, which let you programmatically include these features in your apps. Cloud Client Libraries provide an optimized developer experience by using the natural conventions and styles of each supported language. You also learn about Cloud Code, which will help you develop your Google Cloud applications within integrated development environments.

## Agenda

| | |
|---|---|
| 01 | Cloud APIs and the Google Cloud SDK |
| 02 | Google Cloud CLI |
| 03 | Cloud Client Libraries |
| 04 | Cloud Shell and Cloud Code |

Cloud APIs and the Google Cloud SDK are used to interact with Google services.

## Cloud APIs provide programmatic interfaces to Google Cloud services




- Add features like compute, networking, storage, and machine learning to your applications.
- Make calls through HTTP, JSON, and gRPC interfaces.
- Access sensitive resources by supplying credentials.

Cloud APIs provide programmatic interfaces to Google Cloud services. You can use a Google Cloud resource or service in your application by calling a corresponding Cloud API.

- Cloud APIs let you use powerful features like compute, networking, storage, and machine learning in your applications.
- Cloud APIs can be called by using HTTP requests with JavaScript Object Notation, or JSON, payloads. They can also be called by using Google Remote Procedure Call, or gRPC, requests. gRPC is an open source, remote procedure call framework that can be run anywhere and uses an efficient binary request structure.
- To call Cloud APIs, the caller must supply application credentials. These credentials are validated to ensure that an application is allowed to access your Google Cloud project and resources.

[Cloud APIs: https://cloud.google.com/apis]

## Using the Google Cloud SDK with Google Cloud

- For interacting with Google Cloud services:
  - Command-line tools
  - Language-specific Cloud Client Libraries
- The Google Cloud SDK leverages Cloud APIs.

The Google Cloud SDK is used to interact with Google Cloud products and services.

- The SDK features are in two categories: command-line tools and language-specific Cloud Client Libraries.
- These tools and libraries use Cloud APIs to communicate with Google Cloud.

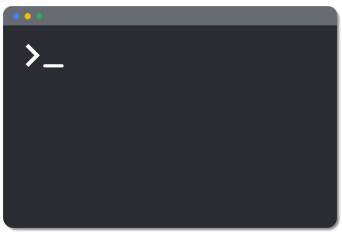
[Google Cloud SDK: https://cloud.google.com/sdk]

## Agenda

| | |
|---|---|
| 01 | Cloud APIs and the Google Cloud SDK |
| 02 | Google Cloud CLI |
| 03 | Cloud Client Libraries |
| 04 | Cloud Shell and Cloud Code |

Next, we look at the Google Cloud CLI.

## Using the Google Cloud CLI (gcloud CLI)




- Provides tools to manage most Google Cloud services.
- Used on the command line or in automated scripts.
- Tools wrap the Cloud APIs.
- Command-line tools include:
  - gcloud
  - gsutil
  - bq

- The Google Cloud Command Line Interface, or gcloud CLI, provides tools to manage Google Cloud services from the command line or in automated scripts.
- These tools provide the functionality of the Cloud APIs in an easy-to-use command-line interface. They automate the process of sending credentials in the Cloud API calls and combine multiple Cloud API calls when required to complete a single common task.
- You can use the gcloud CLI to perform most tasks allowed by the Cloud APIs. For example, you can manage virtual machines or deploy applications to run on Google Cloud.
- The gcloud CLI includes many command-line tools. For example, gcloud interacts with Google Cloud services, gsutil manages Cloud Storage buckets and objects, and bq runs queries and manages data in BigQuery. Other command-line tools can be used to manage Kubernetes, emulate Google Cloud services like Firestore or Pub/Sub, or create infrastructure using Terraform.

[Google Cloud Command Line Interface: https://cloud.google.com/cli]

## The gcloud CLI tool lets you do most common tasks on Google Cloud

```
$ gcloud compute instances list
NAME: example-instance
ZONE: us-central1-a
MACHINE_TYPE: n1-standard-1
PREEMPTIBLE:
INTERNAL_IP: 10.240.95.199
EXTERNAL_IP: 107.167.182.44
STATUS: RUNNING

NAME: instance-2
⋮
```

The gcloud CLI tool lets you perform most common tasks on Google Cloud, including creating and managing resources for many services.

This example command, gcloud compute instances list, shows the Compute Engine virtual machine instances for your project.

[gcloud CLI overview: https://cloud.google.com/sdk/gcloud
gcloud CLI cheat sheet: https://cloud.google.com/sdk/docs/cheatsheet]

## Manage CLI tools

```
$ gcloud components list
Your current Google Cloud CLI version is: 443.0.0
The latest available version is: 444.0.0

Components

Status: Update Available
Name: BigQuery Command Line Tool
ID: bq
Size: 1.6MiB
⋮
```

The gcloud CLI tool also lets you manage the Google Cloud CLI tools.

The command "gcloud components list" describes each of the CLI components. Each component is listed as not installed, update available, or installed at the latest version.

To install a component at the current CLI version, use the "gcloud components install" command. For example, to install the Kubernetes cluster manager kubectl, you'd run the command "gcloud components install kubectl."

You can update your version of the Google Cloud CLI with the command "gcloud components update."

Next, we discuss the gcloud, gsutil, and bq CLI tools.

## Using Cloud Storage CLI tools

- `gsutil` can be used to manage buckets and objects.
- `gcloud storage` performs better than gsutil and has an improved interface.
- `gcloud storage buckets` manages Cloud Storage buckets.
- `gcloud storage objects` manages Cloud Storage objects in buckets.
- To create, move, and delete objects, use the commands cp, mv, and rm.

```
$ gcloud storage cp cloud-storage-logo.png gs://my-bucket
Copying file://cloud-storage-logo.png to gs://my-bucket/cloud-storage-logo.png
  Completed files 1/1 | 2.58KiB/2.58KiB
```

Cloud Storage provides reliable, secure, and highly performant object storage. **gsutil** can be used to create and manage buckets and objects.

**gcloud storage** is now the preferred command line tool for managing Cloud Storage. **gcloud storage** performs better than gsutil, and its usage is similar to other gcloud commands.

**gcloud storage buckets** creates, lists, deletes, and manages access control lists for buckets.

**gcloud storage objects** manages objects and their access control lists.

gcloud storage commands can be used to copy, move, list, and delete objects.

This **gcloud storage** command example copies a file from your local machine into a Cloud Storage bucket.

## bq is a command-line tool for BigQuery

- Manages datasets, tables, and other BigQuery entities.
- Runs queries.

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

Waiting on job_dcda37c0bbed4c669b04dfd567859b90 ... (0s) Current status: DONE
+---------------+-------+
|     word      | count |
+---------------+-------+
| Praising      |   4   |
| raising       |   5   |
| raisins       |   1   |
| praising      |   8   |
| dispraising   |   2   |
| dispraisingly |   1   |
+---------------+-------+
```

bq is a command-line tool for BigQuery, Google Cloud's serverless, highly scalable, and cost-effective data warehouse. bq can be used to manage datasets, tables, and other BigQuery entities, but its primary purpose is running queries.

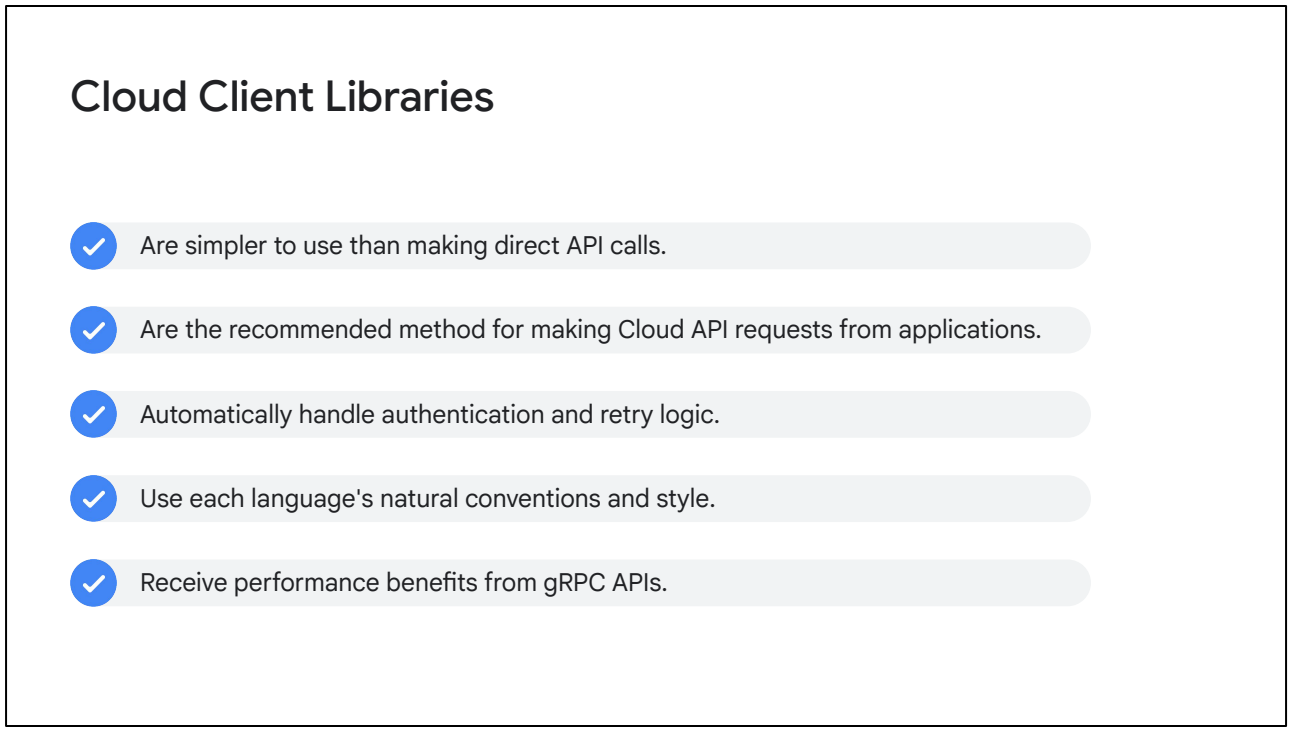
This example query searches for all occurrences of a word in a sample public dataset that contains the complete word index of Shakespeare's works.

[Using the bq command-line tool: https://cloud.google.com/bigquery/bq-command-line-tool]

## Agenda

| | |
|---|---|
| 01 | Cloud APIs and the Google Cloud SDK |
| 02 | Google Cloud CLI |
| 03 | Cloud Client Libraries |
| 04 | Cloud Shell and Cloud Code |

Next, we discuss the Cloud Client Libraries.




- Using a Cloud Client Library is easier than making direct API calls.
- The Cloud Client Libraries are the recommended method for accessing Google Cloud resources from your applications.
- The Cloud Client Libraries provide an optimized developer experience by handling low-level communication with the server, including authentication. They also provide retry logic for transient network failures.
- These libraries use the natural conventions and style for each of the supported languages.
- Many libraries also give you performance benefits by automatically calling the gRPC Cloud APIs.

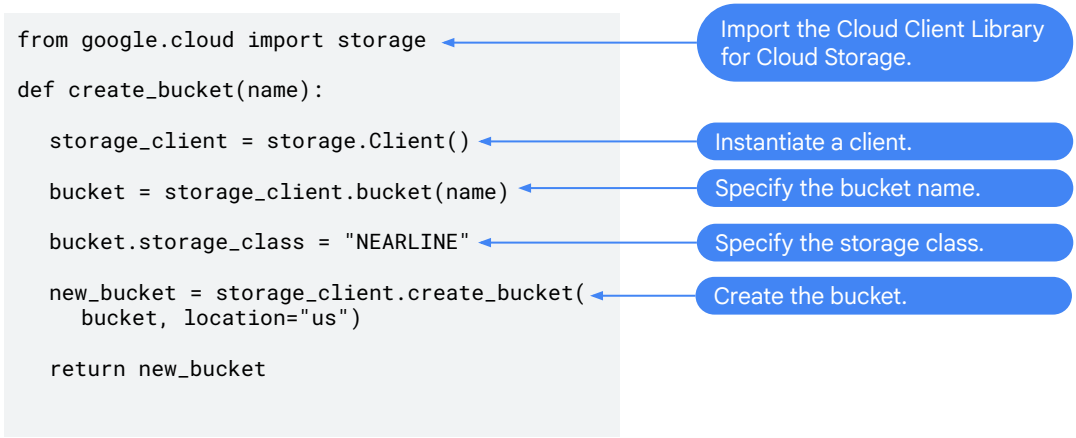
[Cloud Client Libraries: https://cloud.google.com/apis/docs/cloud-client-libraries
Client libraries explained:
https://cloud.google.com/apis/docs/client-libraries-explained#grpc_apis
Authentication overview: https://cloud.google.com/docs/authentication/]

## Cloud Client Libraries are available for many popular programming languages

Cloud Client Libraries are available for many of the most popular programming languages. Supported languages include Python, Node.js, Java, Go, PHP, Ruby, C++, and the .NET languages, including C#.

If your application uses any of these languages, you will probably want to use the corresponding Cloud Client Library.

## Use Python to create a Cloud Storage bucket

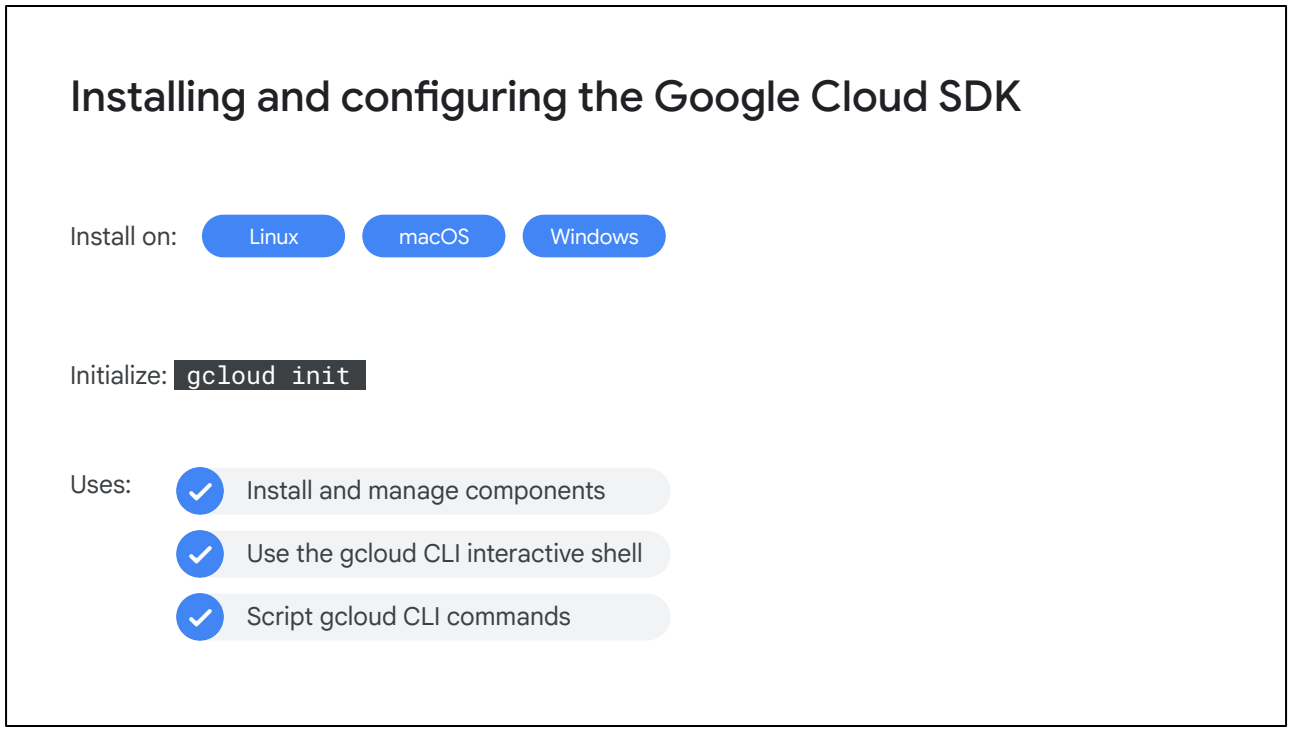



Here's an example of using the Python Cloud Client Library to create a Cloud Storage bucket.

Every package provides a client that interacts with an API. Your application runs with a particular identity, which is typically a service account. This example imports the Cloud Storage client library, instantiates the client by using the default credentials provided by the service account, and creates a cloud bucket.

The Cloud Client Libraries let you easily manage your Google Cloud resources by using the natural style of the language you have chosen.

[Authenticating as a service account: https://developers.google.com/identity/protocols/application-default-credentials
Authentication -- google-api-core documentation: https://googleapis.dev/python/google-api-core/latest/auth.html]




- You can download and install the Google Cloud SDK on Linux, macOS, and Windows.
- The Google Cloud SDK is initialized by running 'gcloud init.' After it's initialized, you can immediately start using the Google Cloud SDK.
- You can install and manage SDK components and use the gcloud CLI interactive shell, which provides prompt completion and suggests command options. You can also script gcloud CLI commands to automate your processes.

[Installing the gcloud CLI: https://cloud.google.com/sdk/downloads]

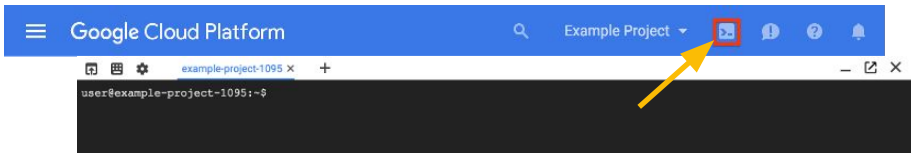
## Agenda

| | |
|---|---|
| 01 | Cloud APIs and the Google Cloud SDK |
| 02 | Google Cloud CLI |
| 03 | Cloud Client Libraries |
| 04 | Cloud Shell and Cloud Code |

Finally, we look at Cloud Shell and Cloud Code.

## Cloud Shell provides command-line access to a free admin VM

- Browser-based access to a temporary virtual machine
  - 5 GB of persistent disk storage
  - The Google Cloud SDK is pre-installed
- Built-in authorization to Google Cloud projects and resources
- Built-in code editor




Cloud Shell is a free admin machine with browser-based command-line access that is used from the Google Cloud console.

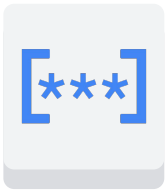
- It provides you with a temporary virtual machine instance that has 5 GB of persistent disk storage. When you start Cloud Shell, it provisions a Compute Engine virtual machine that runs a Debian-based Linux operating system. Cloud Shell instances are provisioned on a per-user, per-session basis. The instances persist only while your Cloud Shell session is active and terminate after an hour of inactivity. When a new instance needs to be provisioned, it retains the persistent disk that was used with the previous instance.

- The Google Cloud SDK comes pre-installed in Cloud Shell with built-in authorization to your Google Cloud projects and resources.

- Cloud Shell comes with a built-in code editor, based on Theia, that you can use to browse directories and view and edit files within your VM.

[Cloud Shell documentation: https://cloud.google.com/shell/docs/
Cloud Shell Editor interface overview:
https://cloud.google.com/shell/docs/editor-overview]

## Use Cloud Code to develop cloud applications in your favorite IDE



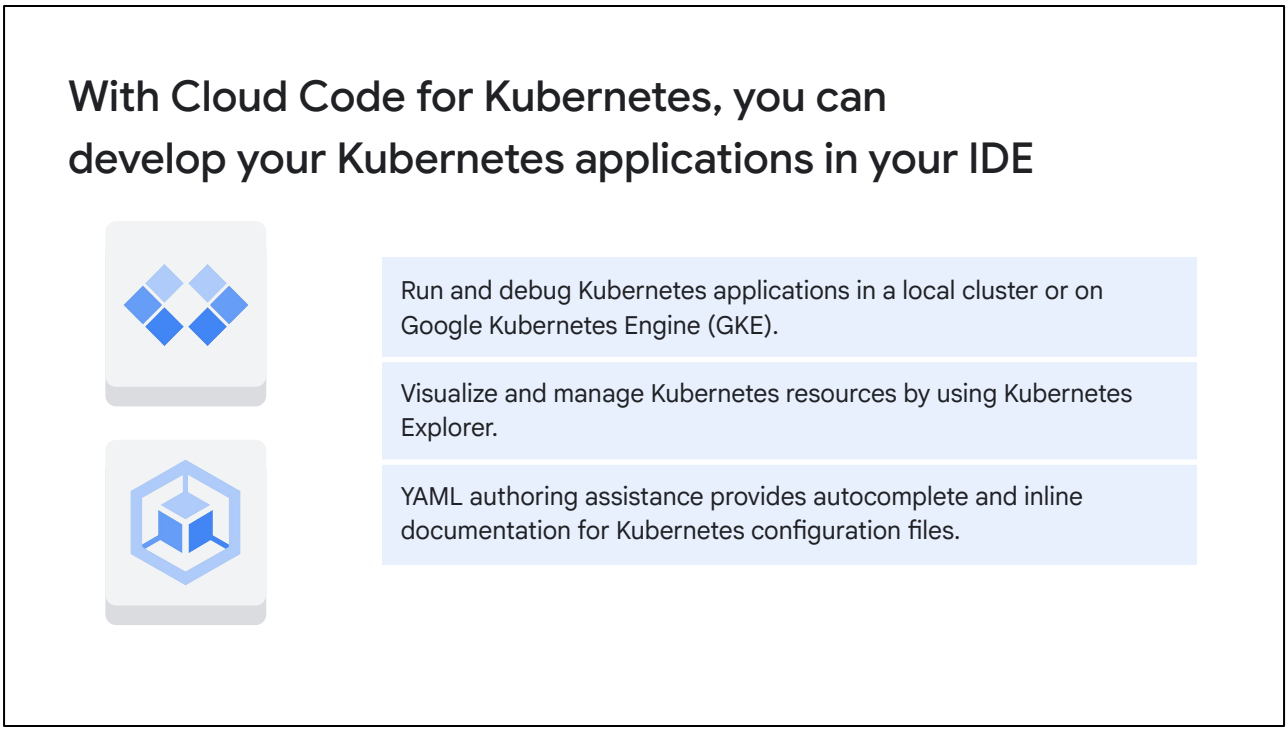

**Cloud Code:**

- Is a set of IDE plugins that make it easier to create, deploy, and debug cloud applications.
- Is available for VSCode, JetBrains IDEs, and Cloud Shell Editor.
- Streamlines common workflows.
- Integrates with Secret Manager to securely store sensitive data.
- Manages Cloud APIs and Cloud Client Libraries.

You can use Cloud Code to help you develop your cloud applications in your favorite integrated development environment, or IDE.

- Cloud Code is a set of IDE plugins that make it easier to create, deploy, and debug cloud applications for Google Cloud.
- Cloud Code is available for the Cloud Shell Editor, Visual Studio Code, and the JetBrains IDEs, which include IntelliJ for Java and PyCharm for Python development.
- Cloud Code streamlines common workflows within the IDE by merging non-trivial tasks into a simple user interface inside your IDE.
- Cloud Code integrates with Secret Manager, which is Google Cloud's service for securely storing passwords, keys, certificates, and other sensitive data. This integration lets you manage your sensitive data within the IDE.
- You can also manage Cloud APIs from the IDE. You can browse the available Cloud APIs and see Cloud Client Library documentation specific to your programming language. You can also find and copy code samples that use the Cloud APIs.

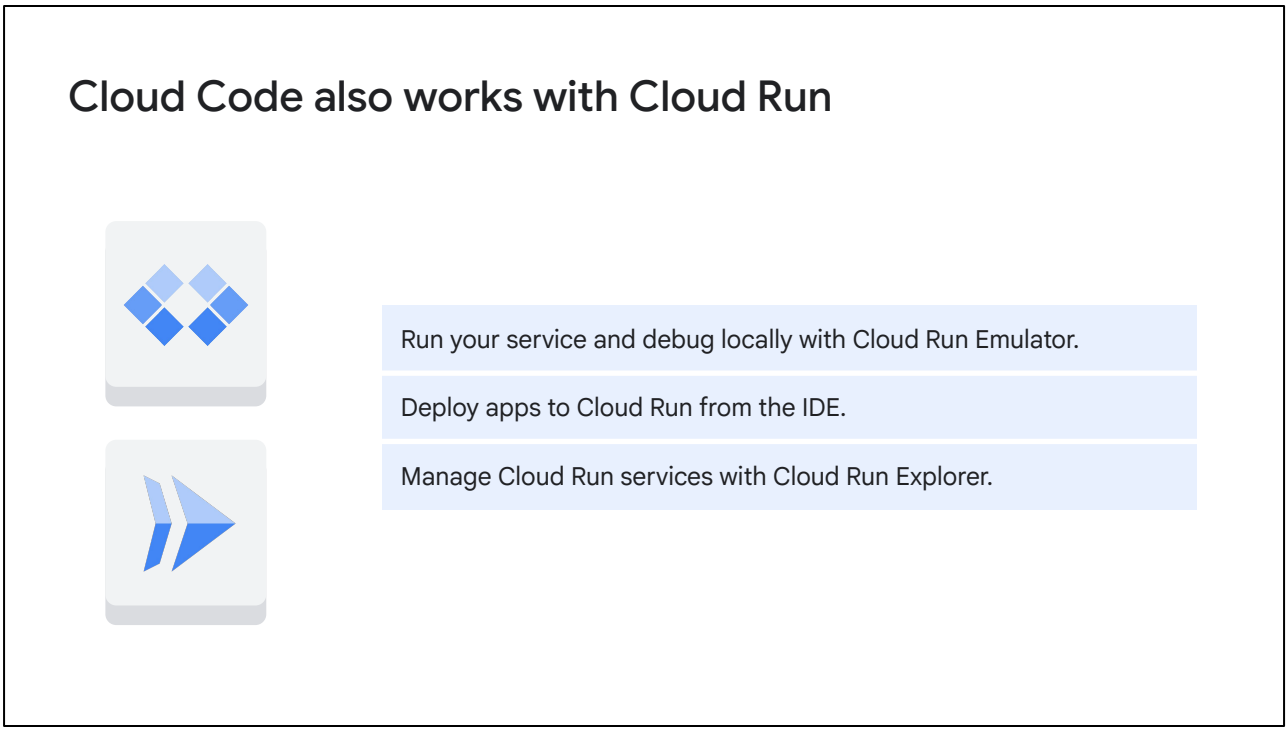
[https://cloud.google.com/code
https://cloud.google.com/secret-manager]




Cloud Code for Kubernetes lets you develop your Kubernetes applications in your IDE.

- You can run and debug your applications in a local cluster or on Google Kubernetes Engine (GKE).
- Cloud Code's Kubernetes Explorer provides you with an easy way to visualize and manage your Kubernetes resources within the IDE. You don't need to remember the associated CLI commands. For example, you can right-click on a pod and stream its logs or open an interactive terminal.
- If you have developed or deployed Kubernetes applications, you know that YAML configuration files for Kubernetes can be complex and require detailed schemas. Cloud Code's YAML authoring assistance simplifies the process of creating and editing these configuration files by providing autocomplete and inline documentation.

[https://cloud.google.com/code/docs/vscode/k8s-overview
https://cloud.google.com/code/docs/vscode/yaml-editing]



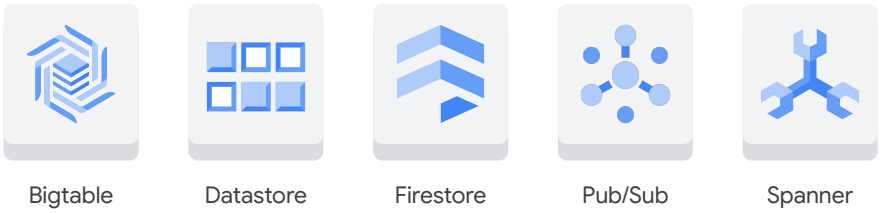

Cloud Code also works with Cloud Run, Google Cloud's fully managed serverless product with autoscaling that scales down to zero.

- You can use Cloud Code to develop your Cloud Run service and then use the Cloud Run Emulator to run and debug it locally.
- When you're ready to deploy your service, you can deploy it from the IDE.
- You can also use Cloud Run Explorer to manage your Cloud Run services from the IDE.

[https://cloud.google.com/code/docs/vscode/cloud-run-overview]

## Use emulators for Google Cloud services

- Use `gcloud beta emulators` to install and manage emulators.
- Switch from using a local emulator to the Google Cloud service without changing application code.
- Develop your applications without consuming project resources.



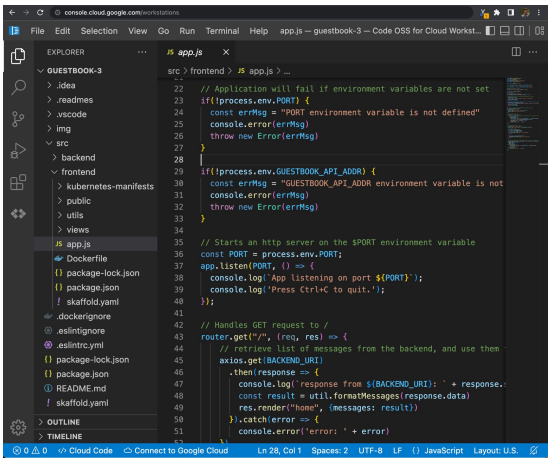

The gcloud CLI includes local emulators for several of the Google Cloud services that you can use in your applications.

- You can use the "gcloud beta emulators" commands to install and manage emulators.
- You do not need to change your application code when you switch between using a local emulator and the Google Cloud service. When you set specified environment variables, Cloud Client Libraries used by your application will automatically connect to the local emulator instead of the Google Cloud service.
- The local emulators let you develop your code without requiring a connection to the corresponding services, and you will not consume project resources.
- Local emulators are currently available for Bigtable, Datastore, Firestore, Pub/Sub, and Spanner.

[gcloud beta emulators: https://cloud.google.com/sdk/gcloud/reference/beta/emulators]

## Use managed development environments with Cloud Workstations

- It provides fully managed and secure cloud-based development environments for Google Cloud.
- Developers can access fast and consistent development environments by using a browser, SSH, or local IDE.
- Administrators can easily provision, scale, manage, and secure development environments.
- IDE and code run inside your VPC.




Cloud Workstations provides fully managed and secure cloud-based development environments for Google Cloud. Instead of requiring your developers to install software and run setup scripts, you can create a workstation configuration that specifies your environment in a reproducible way.

Developers can access fast and consistent development environments anytime and anywhere, using a browser, SSH, or a local IDE. Cloud Workstations supports any code editors and applications that can be run in a container.

IT admins can easily provision, scale, manage, and secure cloud development environments for all developers on the team. Environments are consistent no matter where the developers are located or what type of computer and network they use.

Cloud Workstations runs on ephemeral Compute Engine VMs, and can be started or stopped on demand or when the IDE is idle to improve cost savings. The IDE runs on customer owned VMs and persistent disks inside the customer VPC, ensuring that the developer machines and code are secure.

[Cloud Workstations: https://cloud.google.com/workstations]

## In this module, you learned ...

- The **Google Cloud APIs** provide programmatic interfaces to Google Cloud services.
- The **Google Cloud SDK** provides a simpler interface when used in scripts or on the command line.
- Use the **Cloud Client Libraries** for your chosen programming language to interact with Google Cloud services.

To summarize, here's what we learned in this module:

- The Google Cloud APIs provide programmatic interfaces to Google Cloud services.
- The Google Cloud SDK provides a simpler interface when used in scripts or on the command line.
- When you're ready to write your application, use the Cloud Client Libraries for your chosen programming language to interact with Google Cloud services.

## In this module, you learned ...

- **Cloud Shell** provides a free virtual machine that can be used to manage your Google Cloud projects and resources.
- Use **Cloud Code** to develop your applications in your favorite IDE.


- Cloud Shell provides a free virtual machine that can be used to manage your Google Cloud projects and resources.
- Use Cloud Code to develop your applications in your favorite IDE.