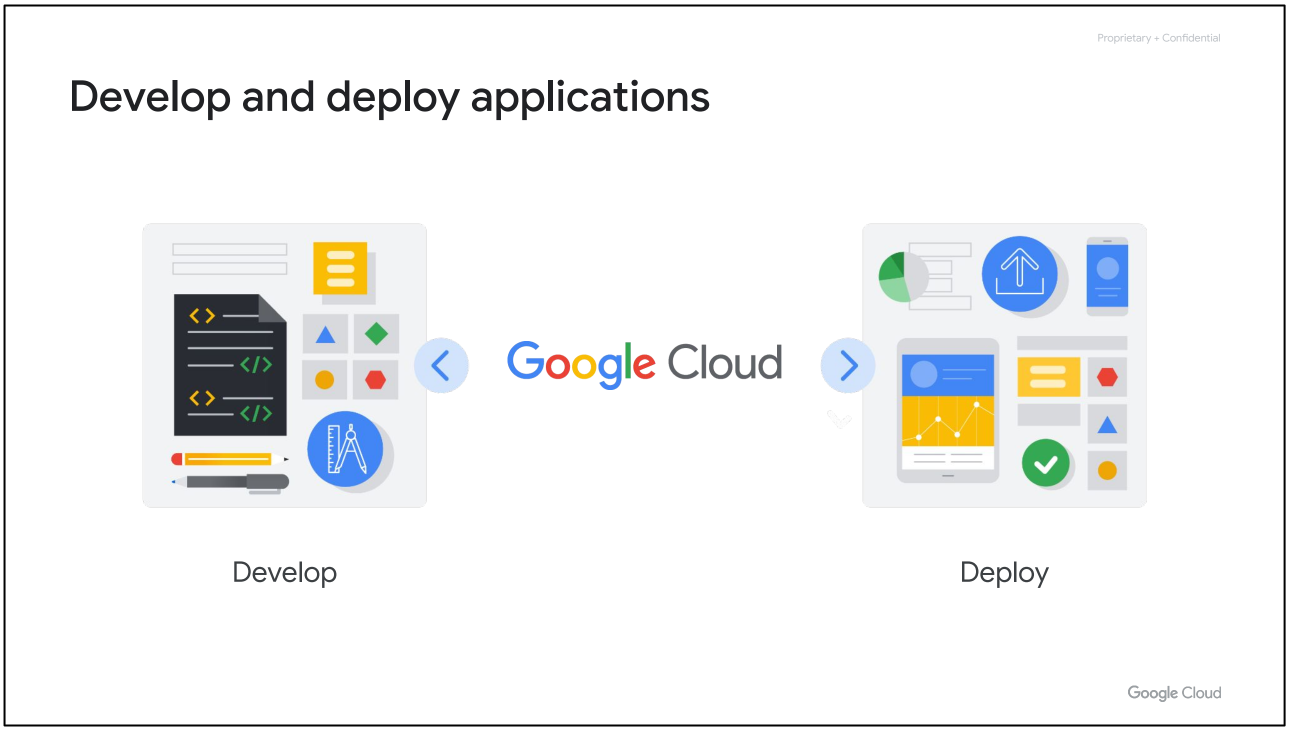
07

Developing and Deploying in the Cloud

Google Cloud

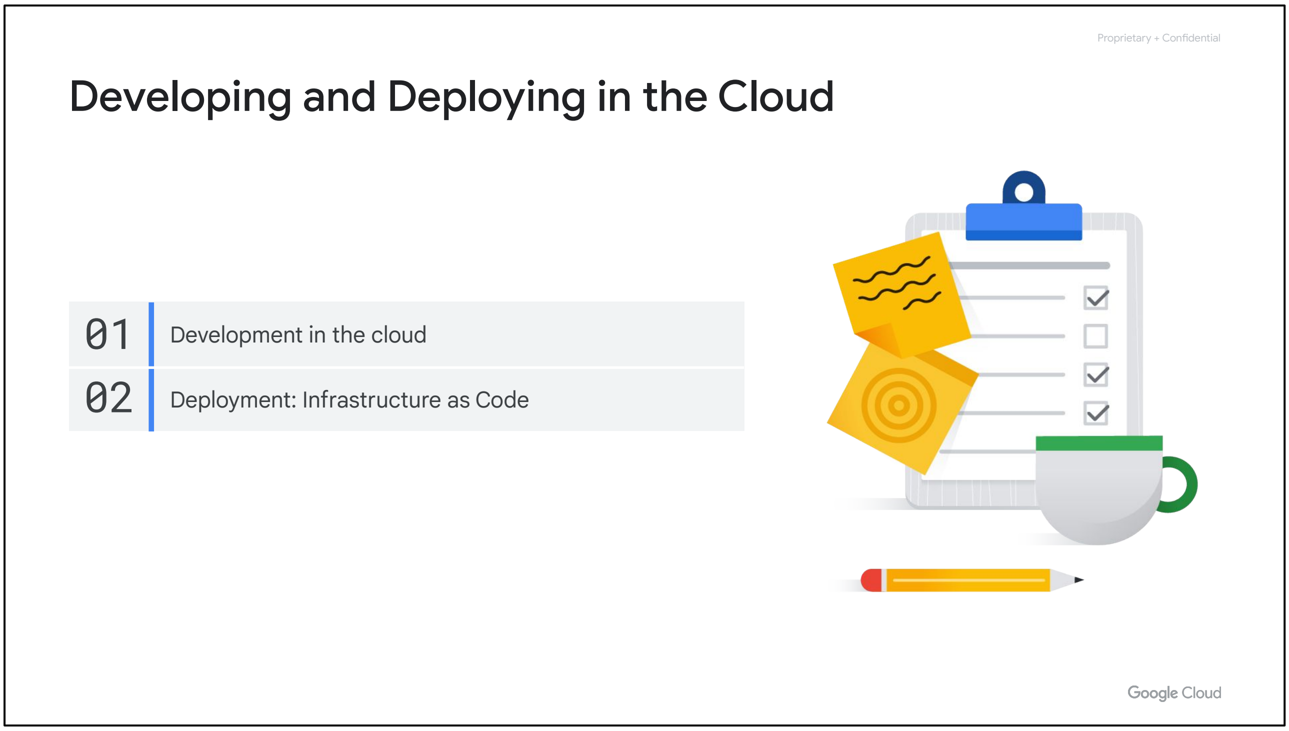


Develop and deploy applications

Develop

Deploy

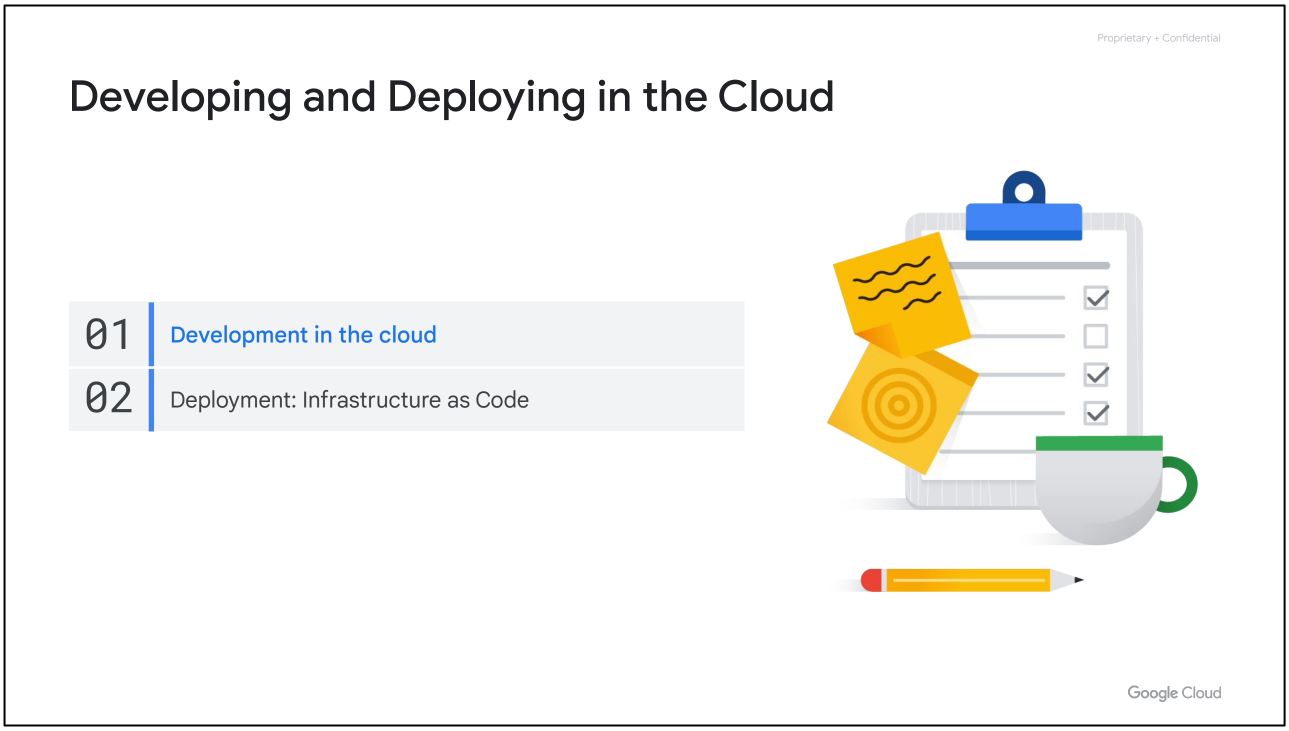
Many users develop impressive applications using Google Cloud's products and services. And when an app is ready, Google Cloud can also be used to deploy it.



Developing and Deploying in the Cloud

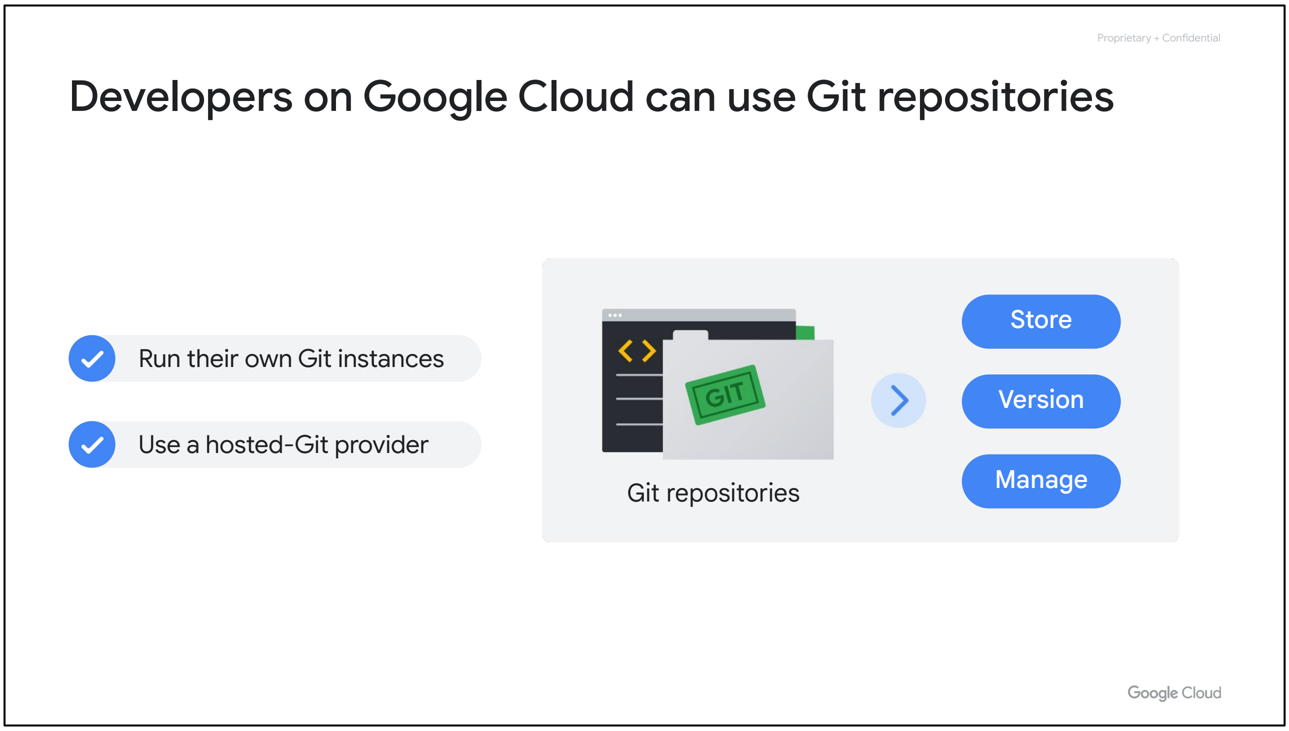
In this section of the course, we'll explore Google Cloud methods for development in the cloud, which includes Cloud Source Repositories, Cloud Functions, and Terraform.

After that, we'll look at deployment with infrastructure as code.



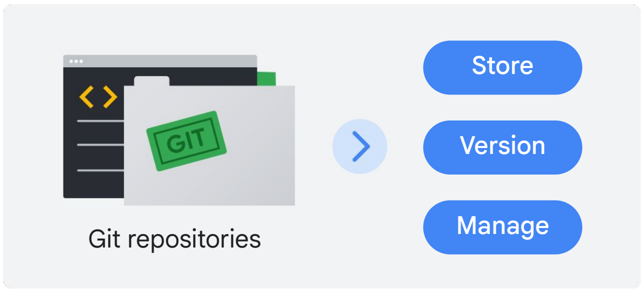
Developing and Deploying in the Cloud

Let's begin with development in the cloud.



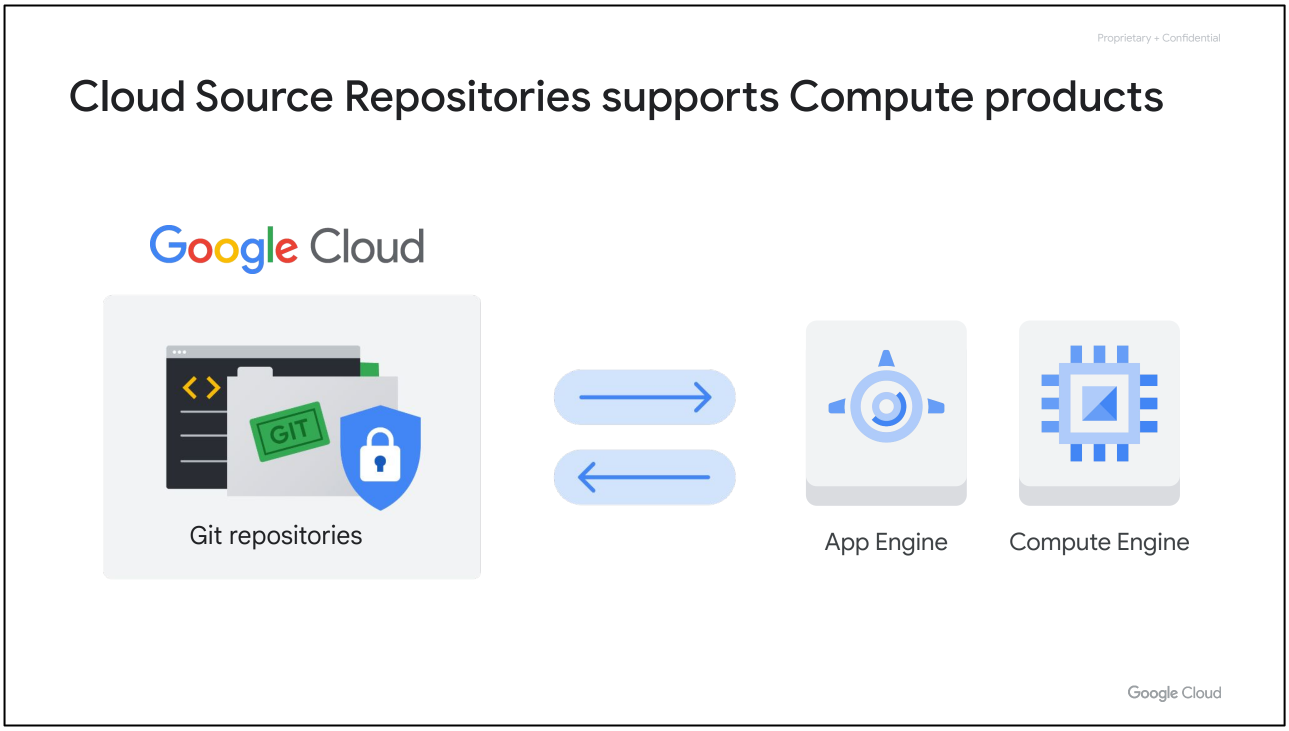
Developers on Google Cloud can use Git repositories

- ✓ Run their own Git instances
- ✓ Use a hosted-Git provider

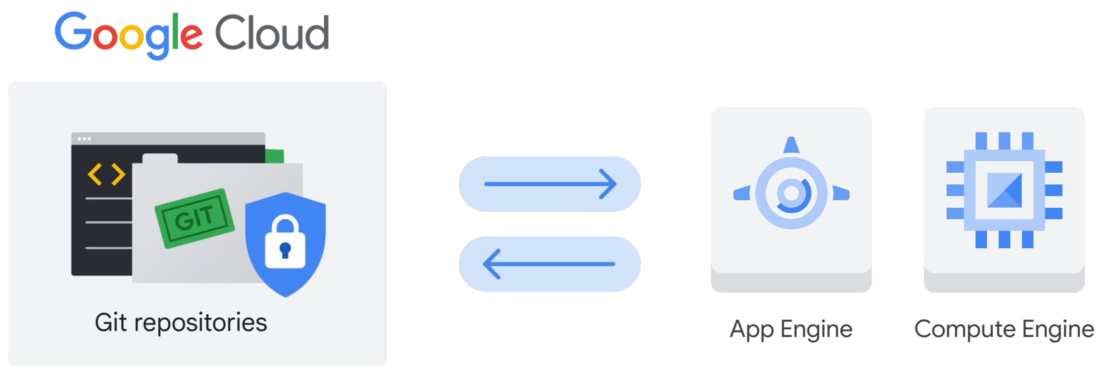


Many Google Cloud customers use Git repositories to store, version, and manage their source code trees. That means they either run their own Git instances, which is a great option if total control is required, or they use a hosted-Git provider, which means less work if total control isn't required.

But what if there were a third option, where you could keep code private to a Google Cloud project and use IAM permissions to protect it, but not have to maintain the Git instance yourself?

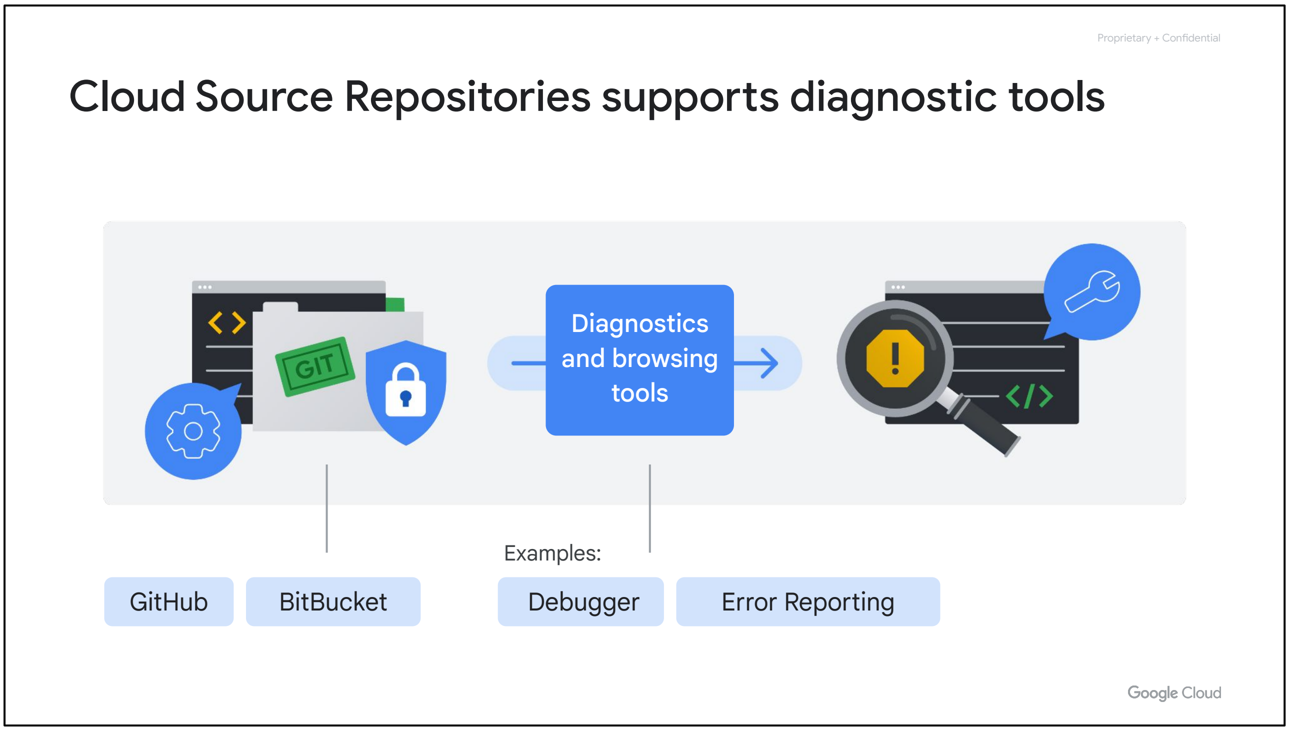


Cloud Source Repositories supports Compute products

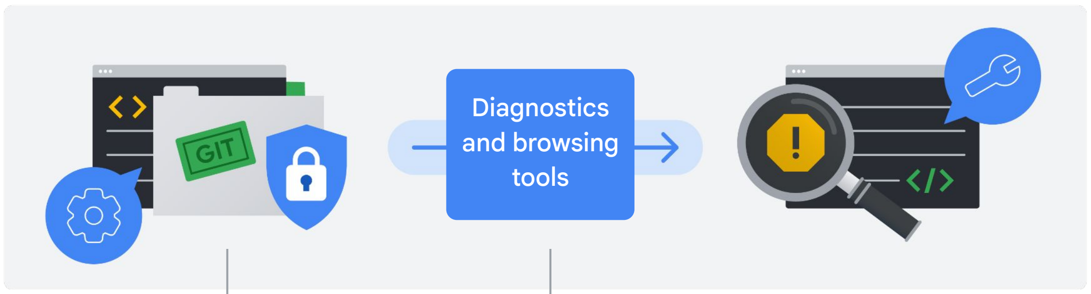


That's available with **Cloud Source Repositories**.

Cloud Source Repositories provides full-featured Git repositories hosted on Google Cloud that support the collaborative development of any application or service, including those that run on App Engine and Compute Engine.



Cloud Source Repositories supports diagnostic tools

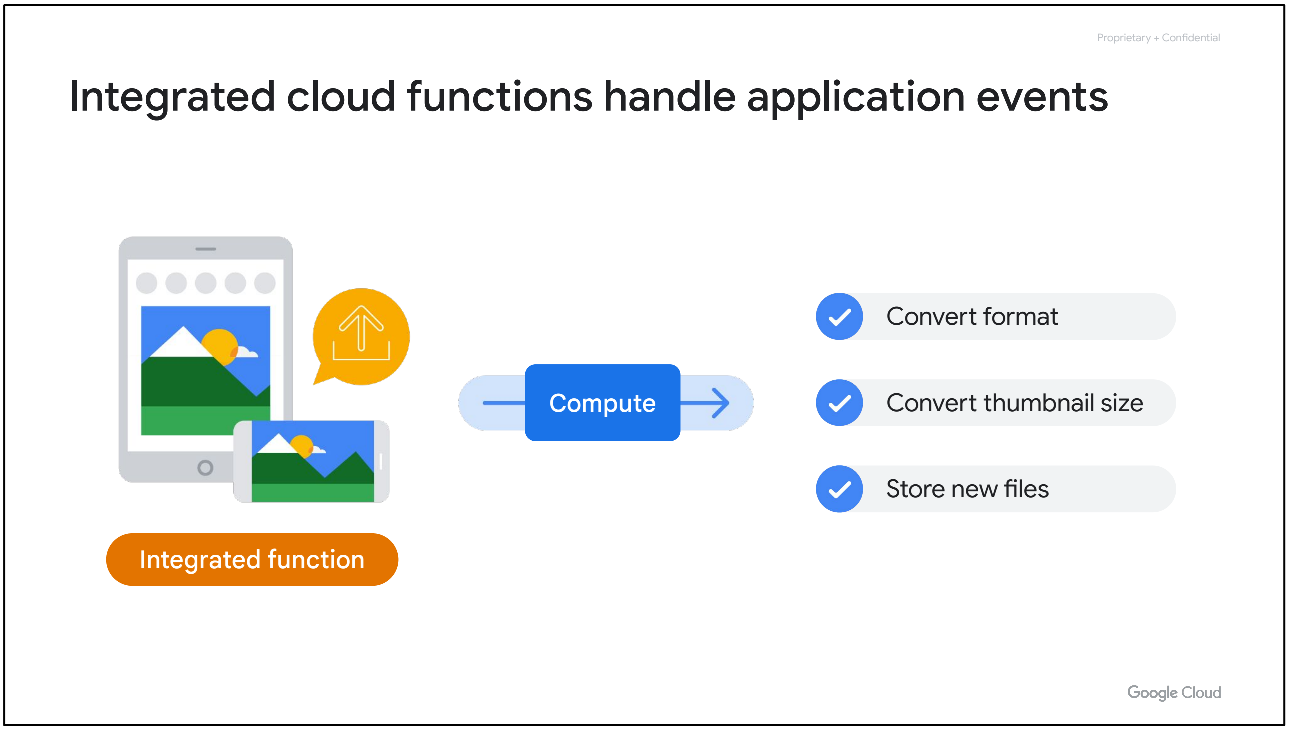


GitHub | BitBucket

Examples: Debugger | Error Reporting

With Cloud Source Repositories, you can have any number of private Git repositories. This allows code associated with a cloud project to be organized the way you choose. It also allows Google Cloud diagnostics tools, like Debugger and Error Reporting, to use the code from Git repositories to track down issues to specific errors in deployed code *without* slowing down your users.

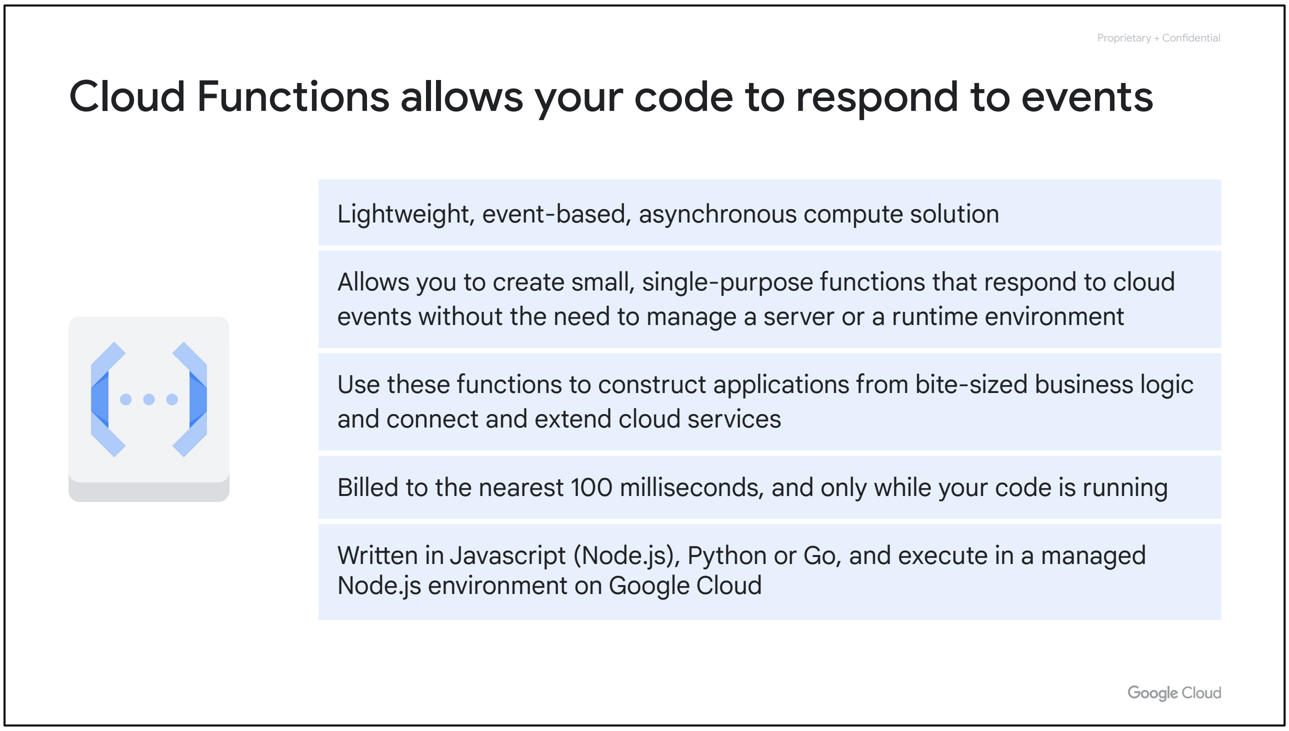
If your code is already in GitHub or BitBucket repositories, it can be migrated into your cloud project and used just like any other repository, including browsing and diagnostics.



Integrated cloud functions handle application events

Many applications contain event-driven parts. For example, maybe you have an application that lets users upload images. When that event takes place, the image might need to be processed in a few different ways, like converting the image to a standard format, converting a thumbnail into different sizes, and storing each new file in a repository.

You could integrate this function into your application, but then you'd have to provide compute resources for it–whether it happens once a millisecond or once a day.



Proprietary + Confidential

Cloud Functions allows your code to respond to events

- Lightweight, event-based, asynchronous compute solution
- Allows you to create small, single-purpose functions that respond to cloud events without the need to manage a server or a runtime environment
- Use these functions to construct applications from bite-sized business logic and connect and extend cloud services
- Billed to the nearest 100 milliseconds, and only while your code is running
- Written in Javascript (Node.js), Python or Go, and execute in a managed Node.js environment on Google Cloud

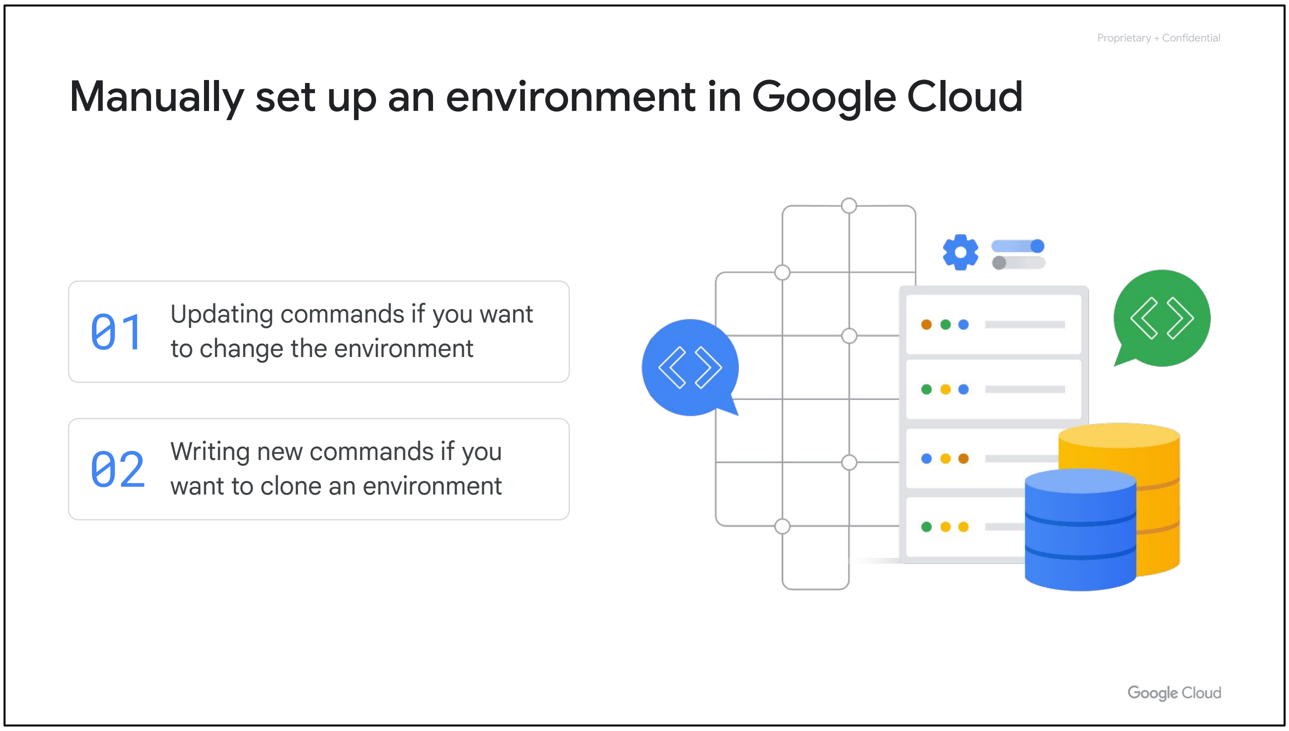
Google Cloud

With **Cloud Functions**, you could write a single-purpose function that completes the necessary image manipulations, and then arrange for it to automatically run whenever a new image is uploaded.

Cloud Functions is a lightweight, event-based, asynchronous compute solution that allows you to create small, single-purpose functions that respond to cloud events without the need to manage a server or a runtime environment. You can use these functions to construct applications from bite-sized business logic. You can also use Cloud Functions to connect and extend cloud services. You are billed to the nearest 100 milliseconds, but only while your code is running.

Individual Cloud Functions are written in Javascript (Node.js), Python, or Go and executed in a managed Node.js environment on Google Cloud. Events from Cloud Storage and Pub/Sub can trigger Cloud Functions asynchronously, or you can use HTTP invocation for synchronous execution.

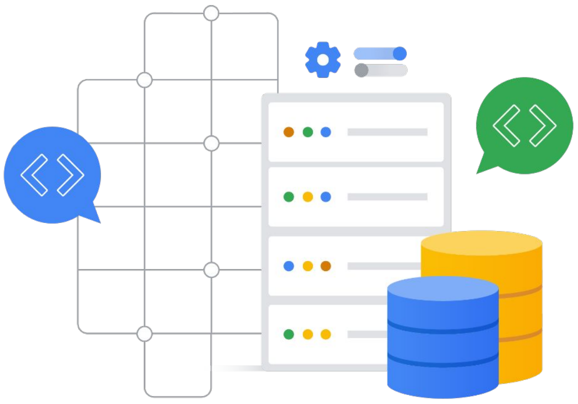
Developing and Deploying in the Cloud



Manually set up an environment in Google Cloud

01 Updating commands if you want to change the environment

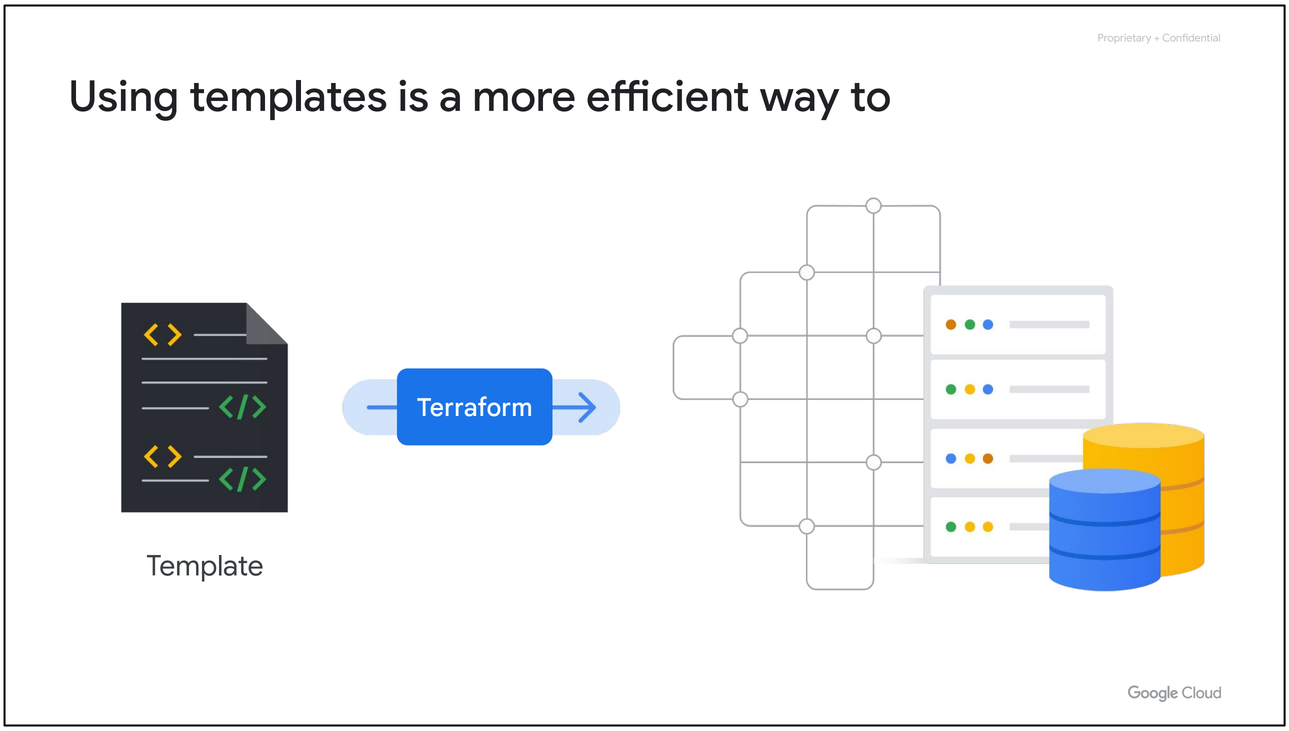
02 Writing new commands if you want to clone an environment



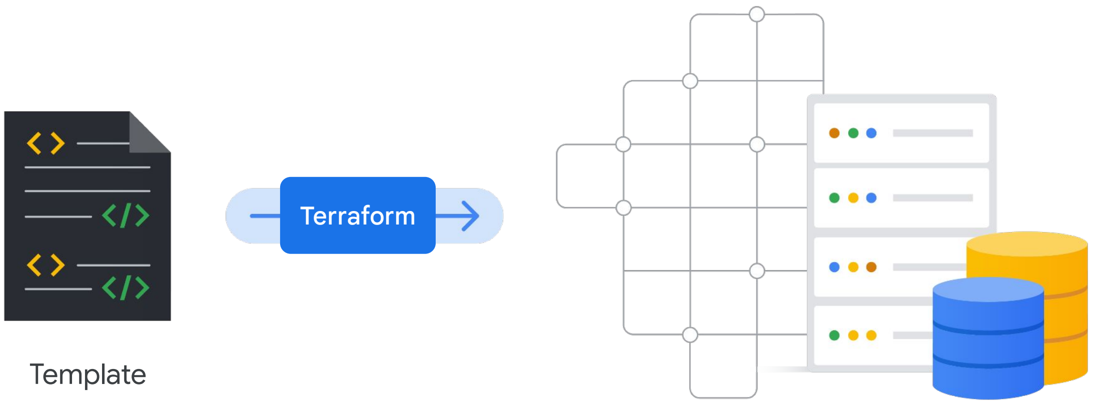
Google Cloud

Creating an environment in Google Cloud can mean lots of work–like setting up a compute network and storage resources, and then keeping track of their configurations.

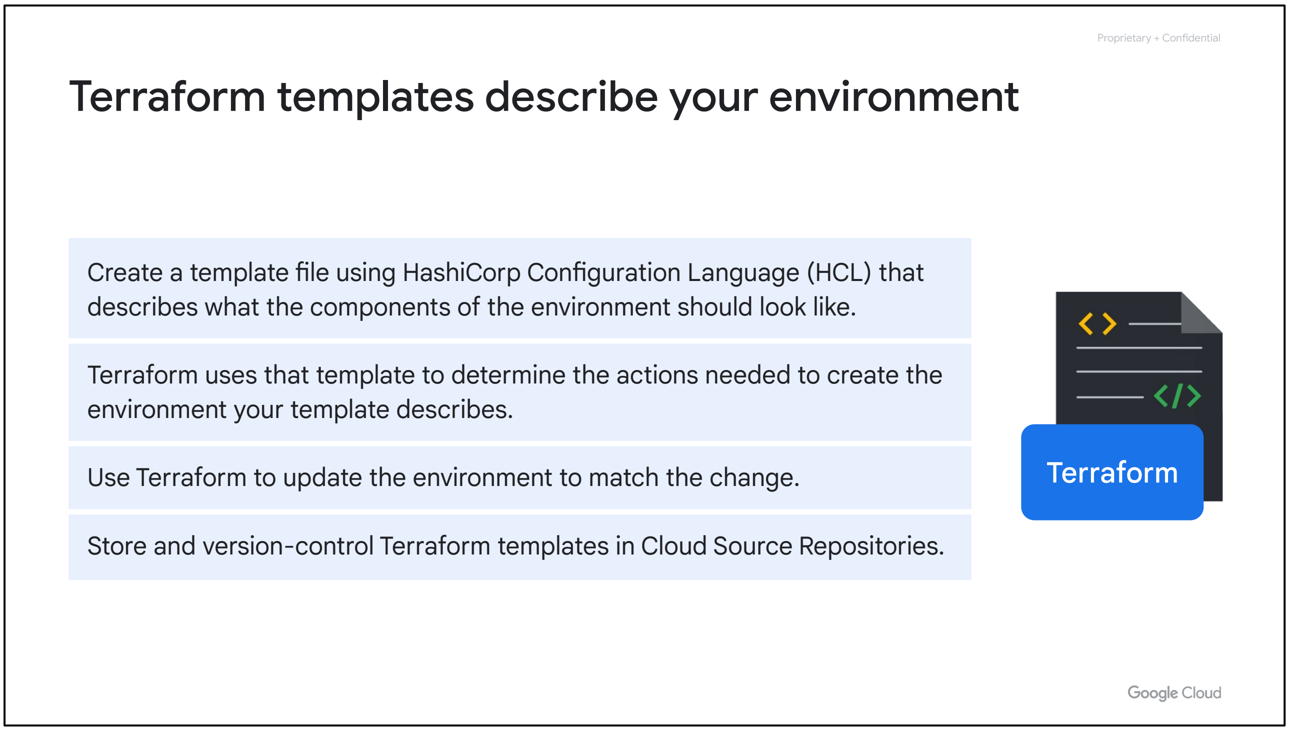
This process can be done manually by writing the commands you need to set up your environment the way you want. However, this is labor intensive, and requires updating commands if you want to change the environment, or manually writing new commands if you want to clone an environment.



Using templates is a more efficient way to



It's more efficient to use a template. Using a template allows you to write the specifications for your application environment in the same way you'd write a configuration file, but your template can then be deployed in a scaled environment to quickly create as many identical application environments as needed. This can be done with **Terraform.**



Terraform templates describe your environment

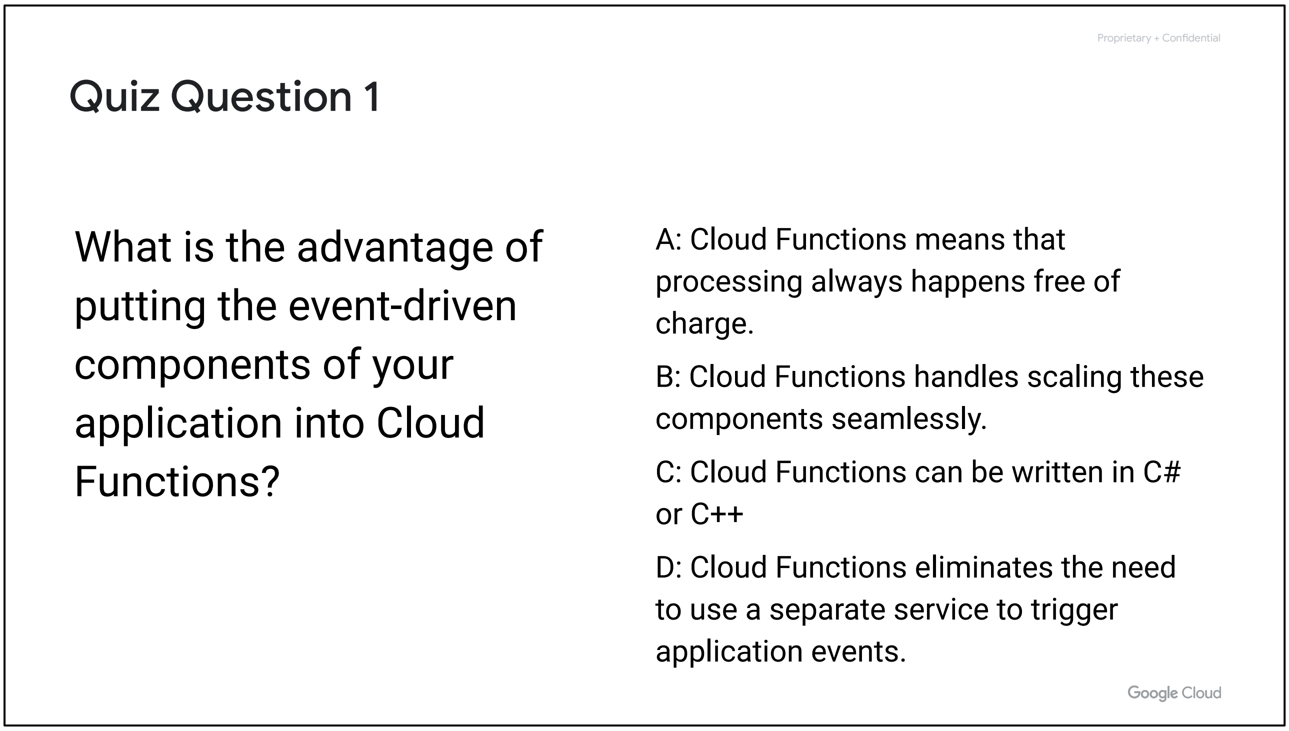
- Create a template file using HashiCorp Configuration Language (HCL) that describes what the components of the environment should look like.
- Terraform uses that template to determine the actions needed to create the environment your template describes.
- Use Terraform to update the environment to match the change.
- Store and version-control Terraform templates in Cloud Source Repositories.

To use Terraform, you create a template file using HashiCorp Configuration Language (HCL) that describes what the components of the environment should look like.

Terraform then uses that template to determine the actions needed to create the environment your template describes. If you need to change the environment, you can edit your template and then use Terraform to update the environment to match the change.

You can store and version-control your Terraform templates in Cloud Source Repositories.

Module Review



Quiz Question 1

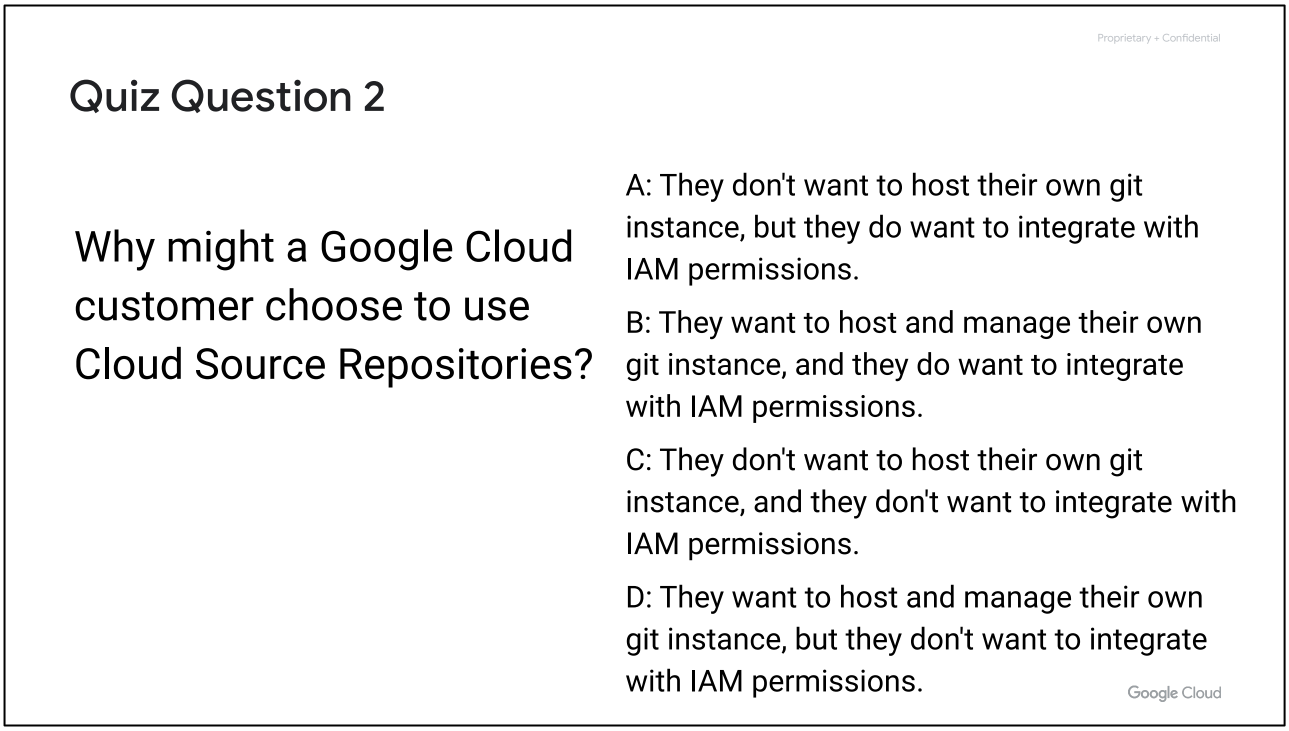
What is the advantage of putting the event-driven components of your application into Cloud Functions?

A: Cloud Functions means that processing always happens free of charge.

B: Cloud Functions handles scaling these components seamlessly.

C: Cloud Functions can be written in C# or C++

D: Cloud Functions eliminates the need to use a separate service to trigger application events.



Quiz Question 2

Why might a Google Cloud customer choose to use Cloud Source Repositories?

A: They don't want to host their own git instance, but they do want to integrate with IAM permissions.

B: They want to host and manage their own git instance, and they do want to integrate with IAM permissions.

C: They don't want to host their own git instance, and they don't want to integrate with IAM permissions.

D: They want to host and manage their own git instance, but they don't want to integrate with IAM permissions.

Quiz Question 3

Why might a Google Cloud customer choose to use Terraform?

A: It is a version control system for your Google Cloud infrastructure layout.

B: It is an infrastructure management system for Kubernetes pods.

C: It enforces maximum resource utilization and spending limits on your Google Cloud resources.

D: It is an infrastructure management system for Google Cloud resources.

Lab

Automating deployment of infrastructure in Google Cloud using Terraform
[60 minutes]

In this lab, you create a Terraform configuration with a module to automate the deployment of Google Cloud infrastructure. Specifically, you deploy one auto mode network with a firewall rule and two VM instances.