Google Cloud

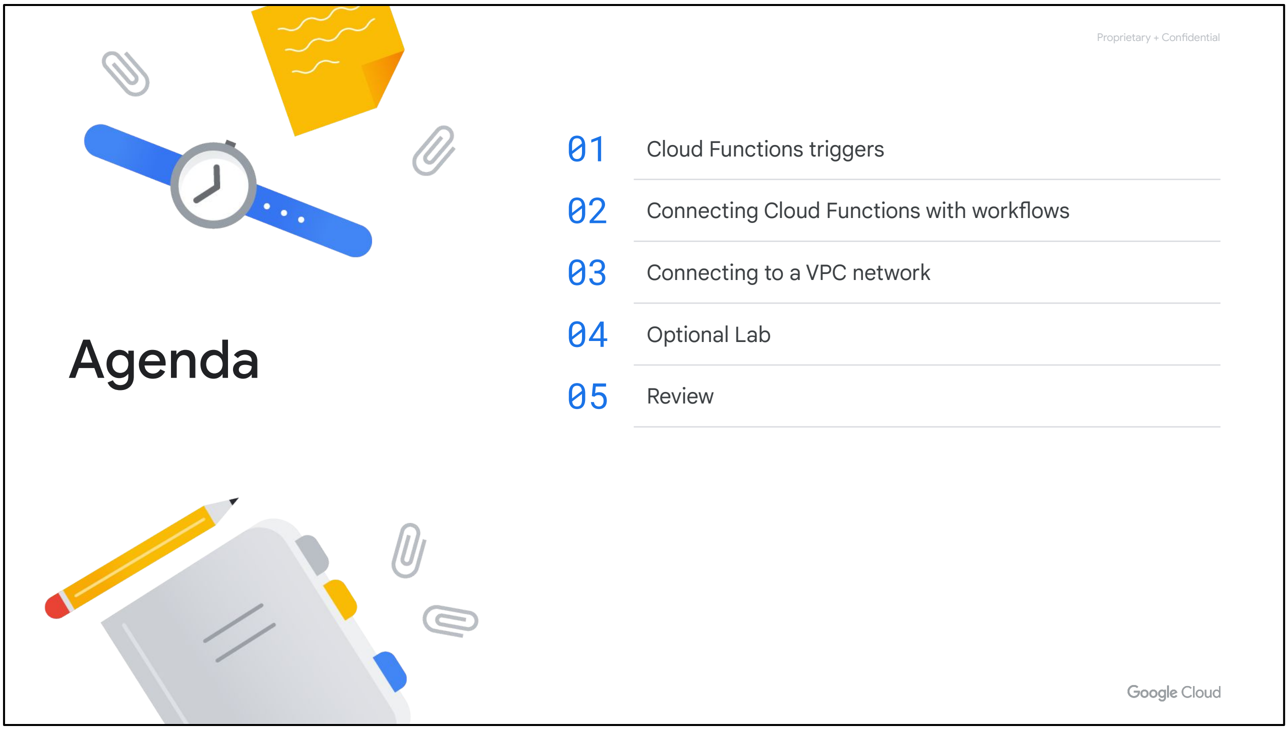
Calling and Connecting Cloud Functions

In this module, we'll discuss how you can call Cloud Functions with function triggers.

You will learn about the different types of function triggers and how to use them when deploying your functions.

We will also discuss how to connect Cloud Functions to Virtual Private Cloud (VPC) networks within your cloud infrastructure.

Finally, we will review connecting Cloud Functions with workflows.



Agenda

01 Cloud Functions triggers

02 Connecting Cloud Functions with workflows

03 Connecting to a VPC network

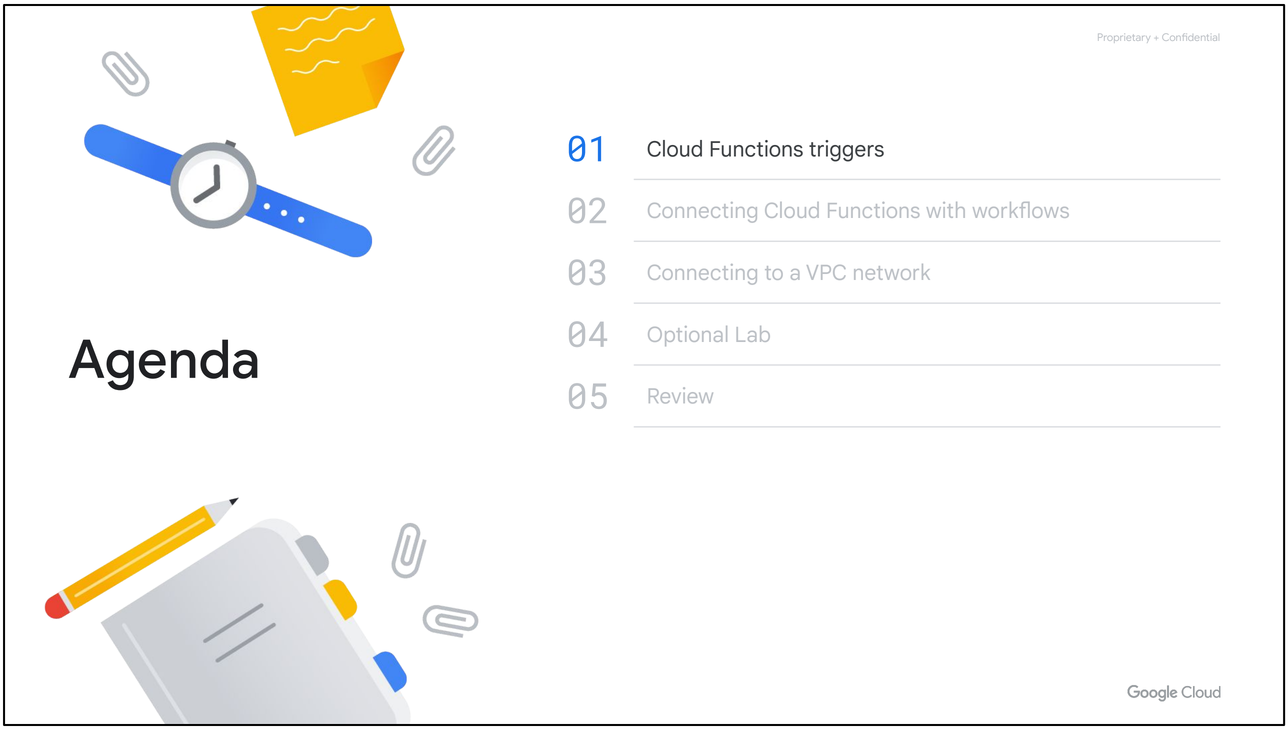
04 Optional Lab

05 Review

We'll first discuss Cloud Functions triggers, then, how to connect Cloud Functions with workflows, and how to connect Cloud Functions to resources in a VPC network.

You can then put this in practice by completing an optional lab to connect Cloud Functions to resources in a VPC network.

Finally, we'll review the topics that were discussed in this module.



Agenda

01 Cloud Functions triggers

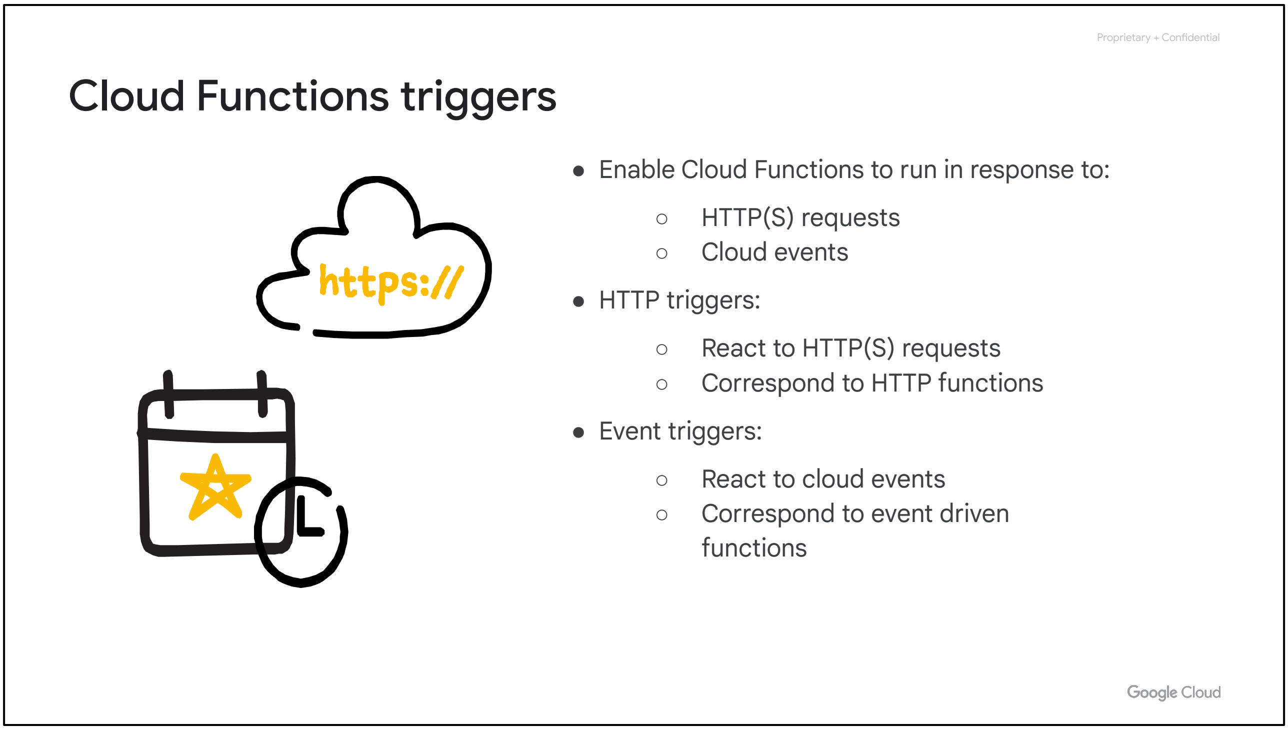
02 Connecting Cloud Functions with workflows

03 Connecting to a VPC network

04 Optional Lab

05 Review

Let’s get started with Cloud Functions triggers.



Proprietary + Confidential

Cloud Functions triggers

- Enable Cloud Functions to run in response to:
  - HTTP(S) requests
  - Cloud events
- HTTP triggers:
  - React to HTTP(S) requests
  - Correspond to HTTP functions
- Event triggers:
  - React to cloud events
  - Correspond to event driven functions

Google Cloud

You set up Cloud Functions to execute in response to various scenarios by specifying a *trigger* for your function.

Triggers can be HTTP(S) requests or one of several supported cloud events.

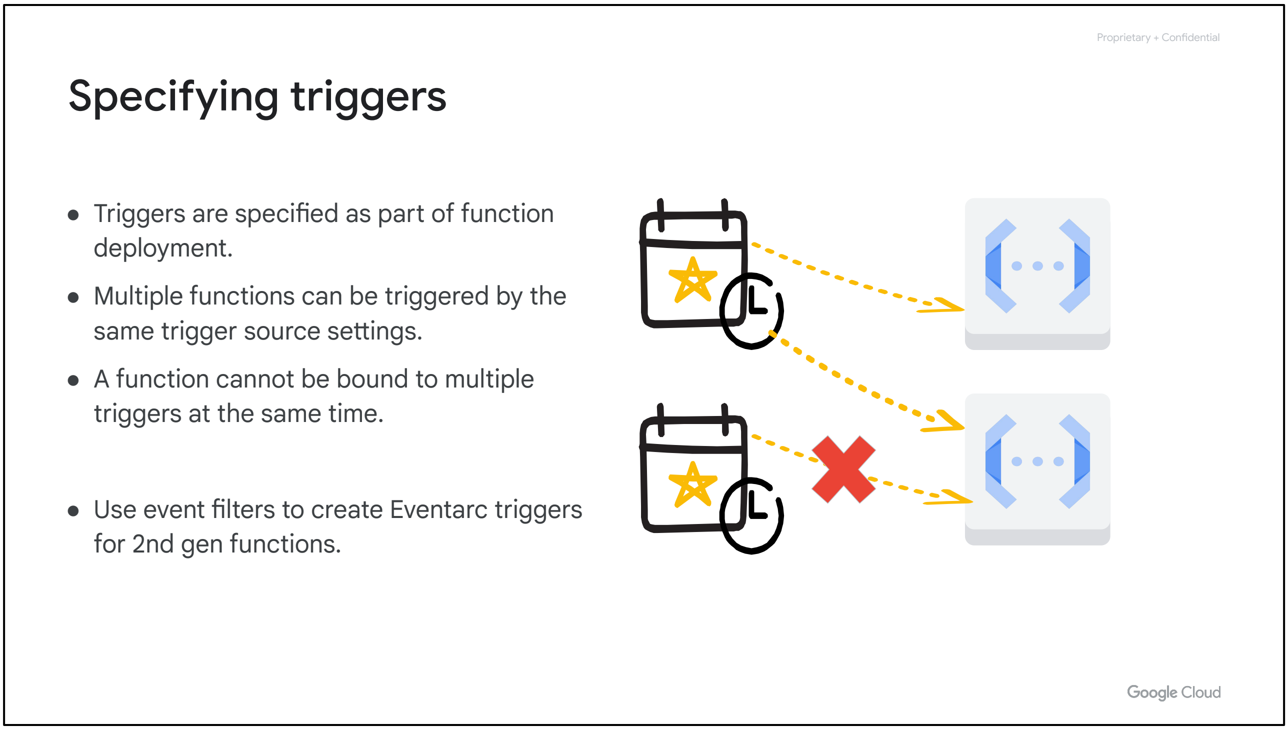
There are two categories of triggers:

- HTTP triggers, which react to HTTP(S) requests, and correspond to HTTP functions.

and,

- Event triggers, which react to events within your Google Cloud project, and correspond to event-driven functions.

Slide graphic from Sketch notes:
https://cloud.google.com/blog/topics/developers-practitioners/learn-cloud-functions-snap



Proprietary + Confidential

Specifying triggers

- Triggers are specified as part of function deployment.
- Multiple functions can be triggered by the same trigger source settings.
- A function cannot be bound to multiple triggers at the same time.

- Use event filters to create Eventarc triggers for 2nd gen functions.

Google Cloud

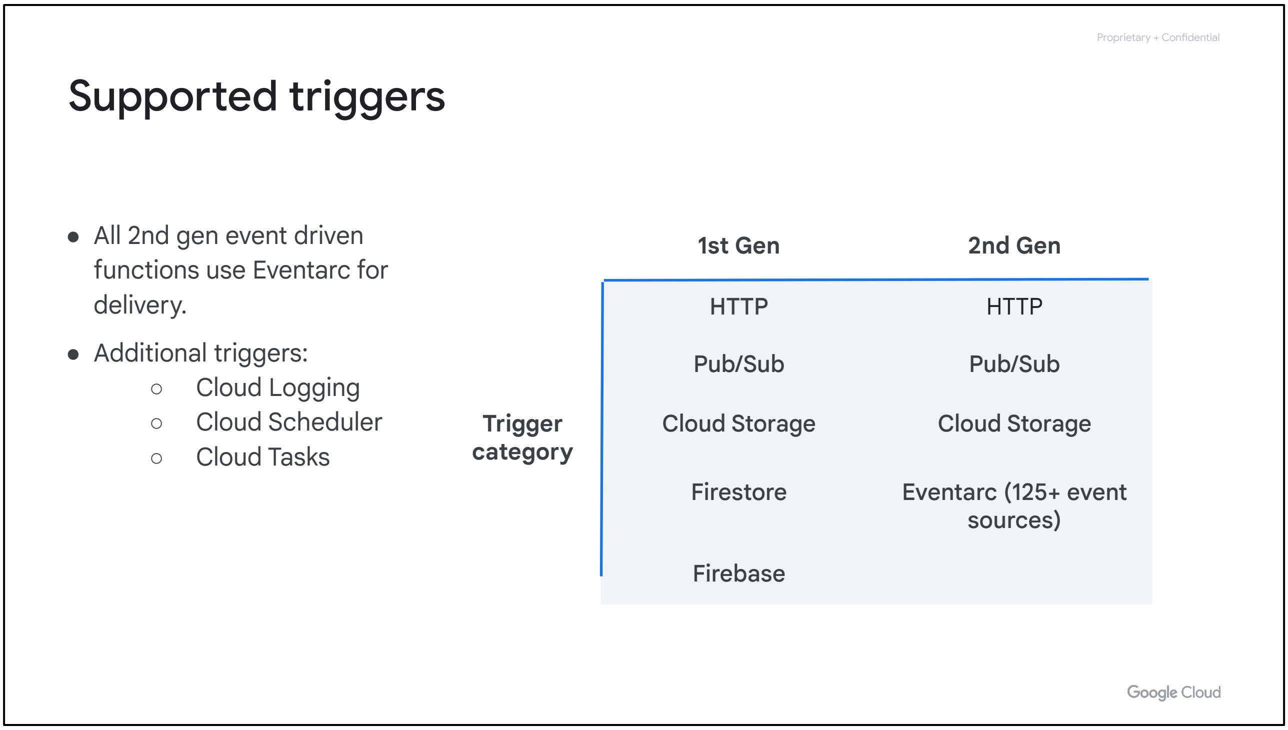
You specify triggers as part of function deployment.

You can have the same event cause multiple functions to execute by deploying multiple functions with the same trigger source settings.

But, you cannot bind the same function to more than one trigger at a time.

With 2nd gen event-driven Cloud Functions, you can create Eventarc event triggers using filters.

Event filters can include the service name, method name, event type, and other information. You can create triggers with the correct event filters in the Google Cloud console or with the gcloud CLI.



Supported triggers

- All 2nd gen event driven functions use Eventarc for delivery.
- Additional triggers:
  - Cloud Logging
  - Cloud Scheduler
  - Cloud Tasks

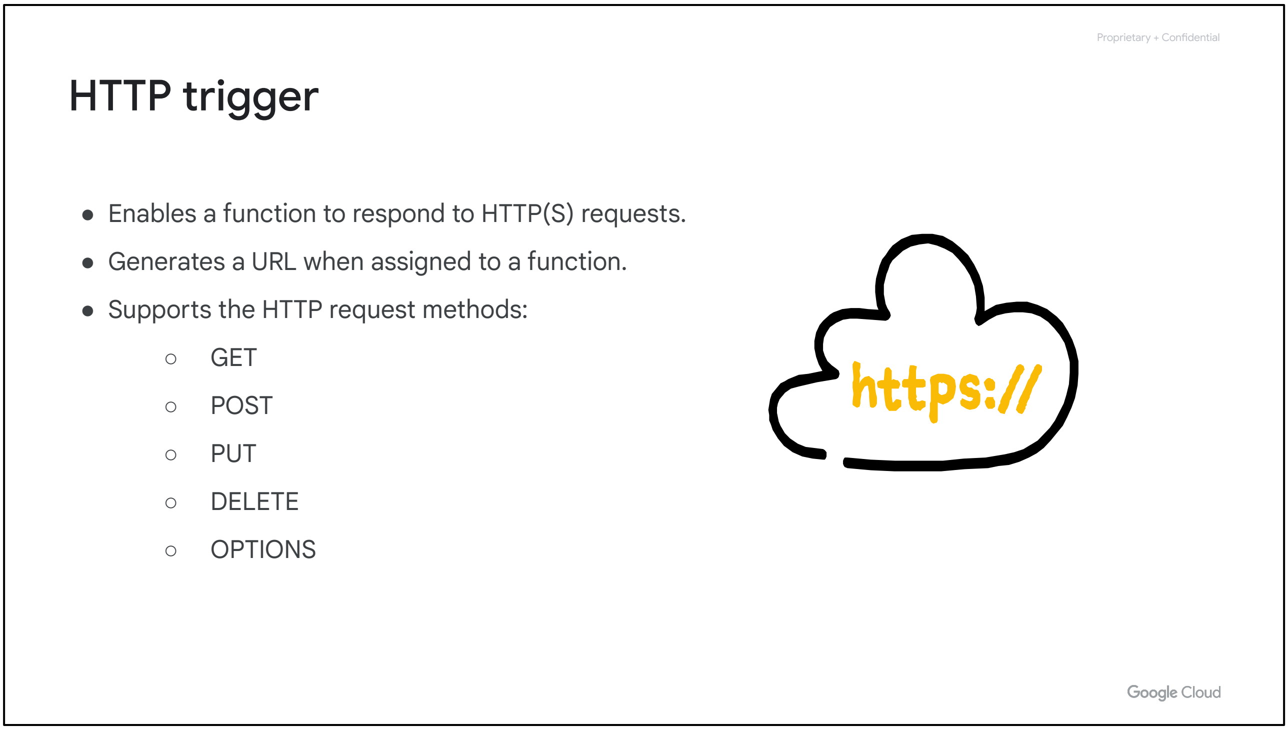
	1st Gen	2nd Gen
Trigger category	HTTP	HTTP
	Pub/Sub	Pub/Sub
	Cloud Storage	Cloud Storage
	Firestore	Eventarc (125+ event sources)
	Firebase	

All event driven functions in 2nd gen Cloud Functions use Eventarc for event delivery.

Eventarc supports more than 125 Google Cloud sources, including events from Cloud Audit Logs, external SaaS event sources, and custom sources by publishing to Pub/Sub.

You can integrate Cloud Functions with any other Google service that supports Pub/Sub as an event bus, for example Cloud Logging and Cloud Scheduler. This is possible because Cloud Functions can be triggered by messages on a Pub/Sub topic.

You can also use HTTP Cloud Functions as task handlers with Cloud Tasks.



HTTP trigger

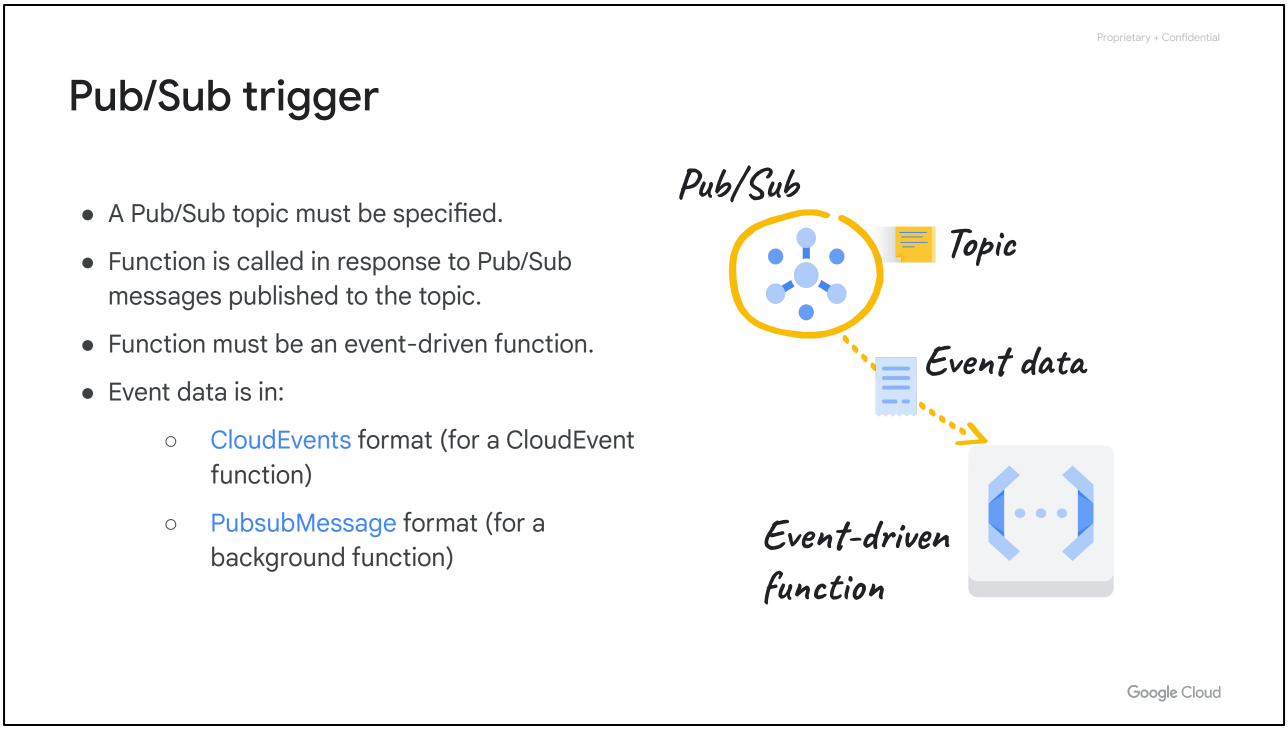
- Enables a function to respond to HTTP(S) requests.
- Generates a URL when assigned to a function.
- Supports the HTTP request methods:
  - GET
  - POST
  - PUT
  - DELETE
  - OPTIONS

HTTP triggers enable a function to run in response to HTTP(S) requests.

When you specify an HTTP trigger for a function, the function is assigned a URL at which it can receive requests.

2nd gen functions always generate a URL for HTTP and Event triggers.

HTTP triggers support the GET, POST, PUT, DELETE, and OPTIONS request methods.



Proprietary + Confidential

Pub/Sub trigger

- A Pub/Sub topic must be specified.
- Function is called in response to Pub/Sub messages published to the topic.
- Function must be an event-driven function.
- Event data is in:
  - CloudEvents format (for a CloudEvent function)
  - PubsubMessage format (for a background function)

Google Cloud

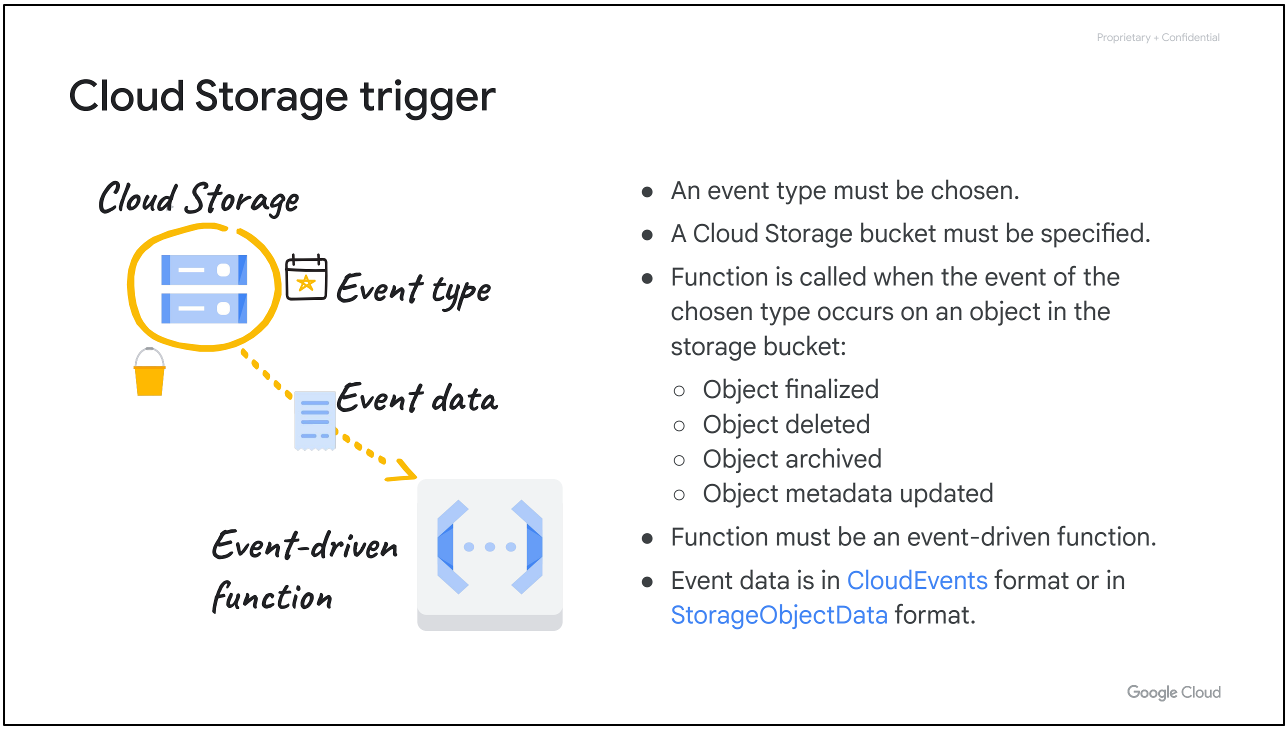
Pub/Sub triggers enable Cloud Functions to be called in response to Pub/Sub messages. When you specify a Pub/Sub trigger for a function, you also specify a Pub/Sub topic. Your function will be called whenever a message is published to the specified topic.

For a function to use a Pub/Sub trigger, it must be implemented as an event-driven function.

If a CloudEvent function is used, the Pub/Sub event data is passed to the function in the CloudEvents format.

If a Background function is used, the Pub/Sub event data is passed to the function in the PubsubMessage format.

In 2nd gen Cloud Functions, Pub/Sub triggers are implemented as a type of Eventarc trigger.



Proprietary + Confidential

Cloud Storage trigger

- An event type must be chosen.
- A Cloud Storage bucket must be specified.
- Function is called when the event of the chosen type occurs on an object in the storage bucket:
  - Object finalized
  - Object deleted
  - Object archived
  - Object metadata updated
- Function must be an event-driven function.
- Event data is in CloudEvents format or in StorageObjectData format.

Google Cloud

Cloud Storage triggers enable a function to be called in response to changes in Cloud Storage.

When you specify a Cloud Storage trigger for a function, you choose an event type and provide a specific Cloud Storage bucket.

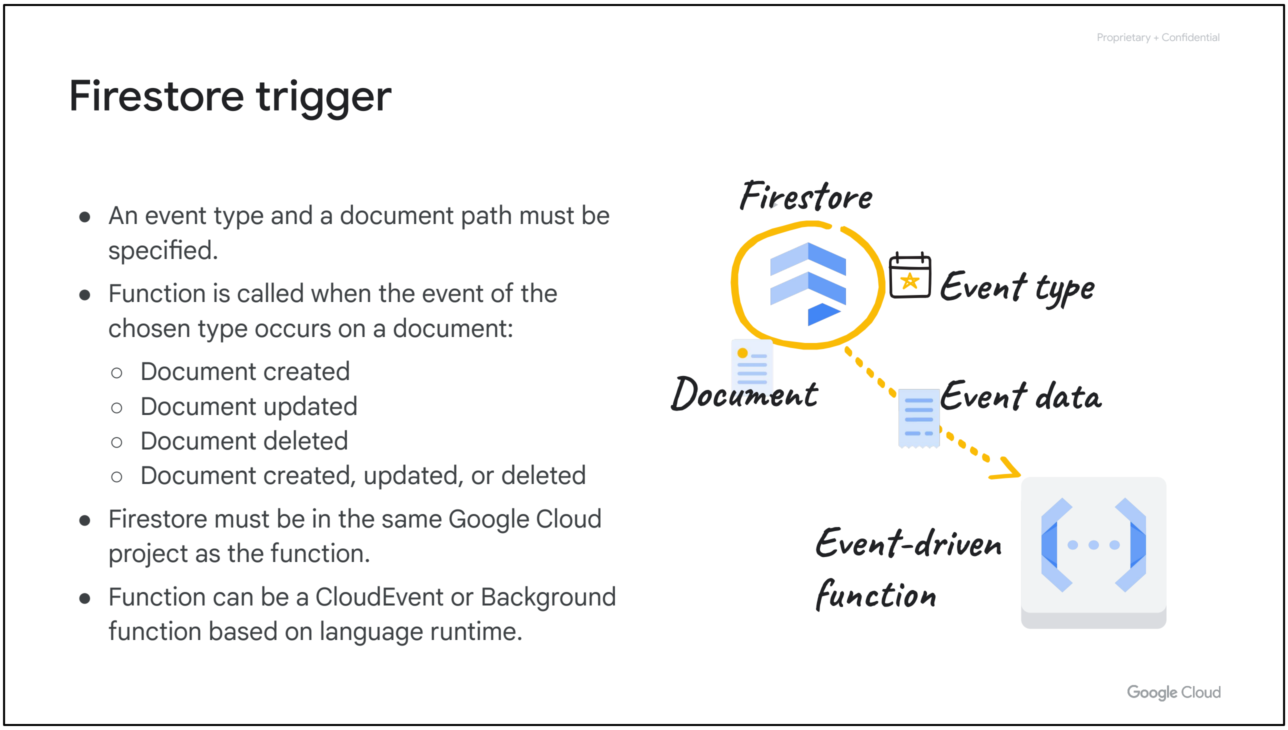
The function will be called whenever a change occurs on an object (file) within the specified bucket.

For a function to use a Cloud Storage trigger, it must be implemented as an event-driven function.

If a CloudEvent function is used, the Cloud Storage event data is passed to the function in the CloudEvents format.

If a Background function is used, the Cloud Storage event data is passed to the function in the StorageObjectData format.

In 2nd gen Cloud Functions, Cloud Storage triggers are implemented as a type of Eventarc trigger.



Firestore trigger

- An event type and a document path must be specified.
- Function is called when the event of the chosen type occurs on a document:
  - Document created
  - Document updated
  - Document deleted
  - Document created, updated, or deleted
- Firestore must be in the same Google Cloud project as the function.
- Function can be a CloudEvent or Background function based on language runtime.

Firestore triggers enable a function to handle events in Firestore that is in the same Google Cloud project as the function.

When you specify a Firestore trigger for a function, you choose an event type, and a document path.

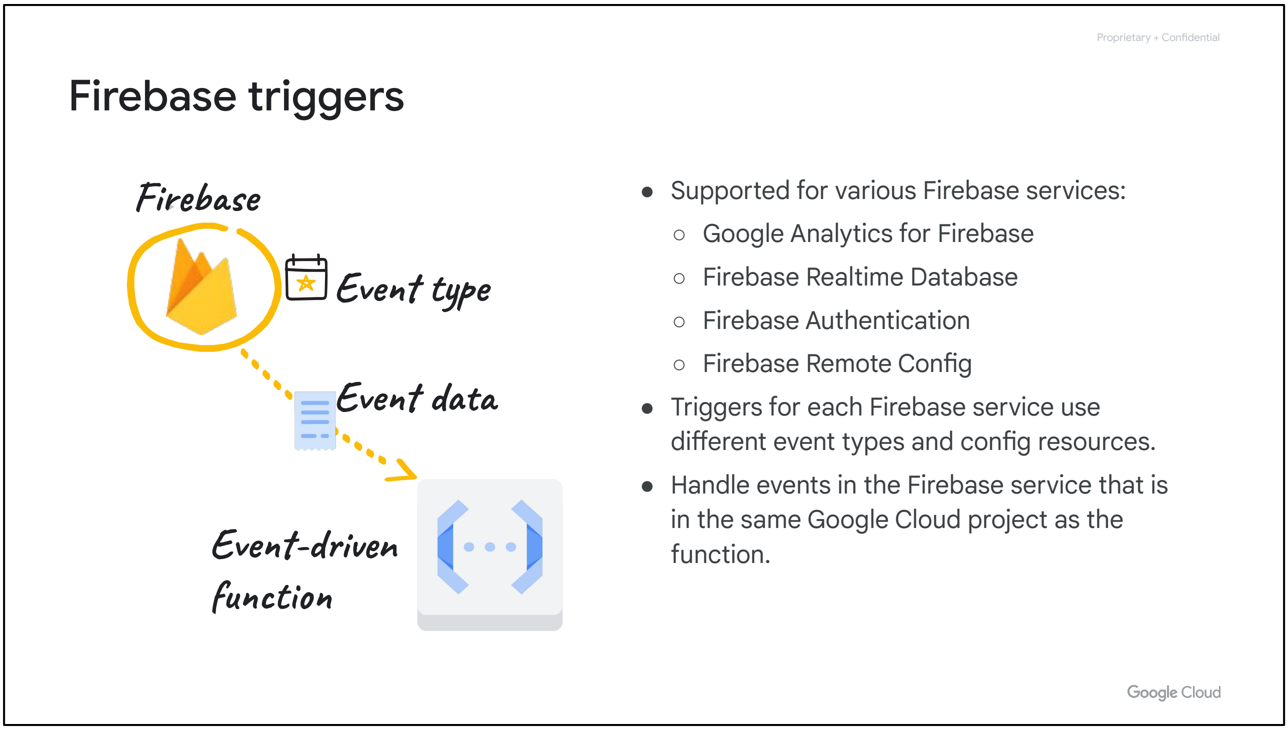
When an event of the specified type occurs on a document, the function is invoked. Firestore supports create, update, delete, and write events.

The function receives a data object with a snapshot of the affected document.

Firestore triggers only apply at the document level. It is not possible to create a trigger for a specific field or collection in the document.

Firestore must be in the same Google Cloud project as the function.

The function can be a CloudEvent or Background function based on the language runtime.



Firebase triggers

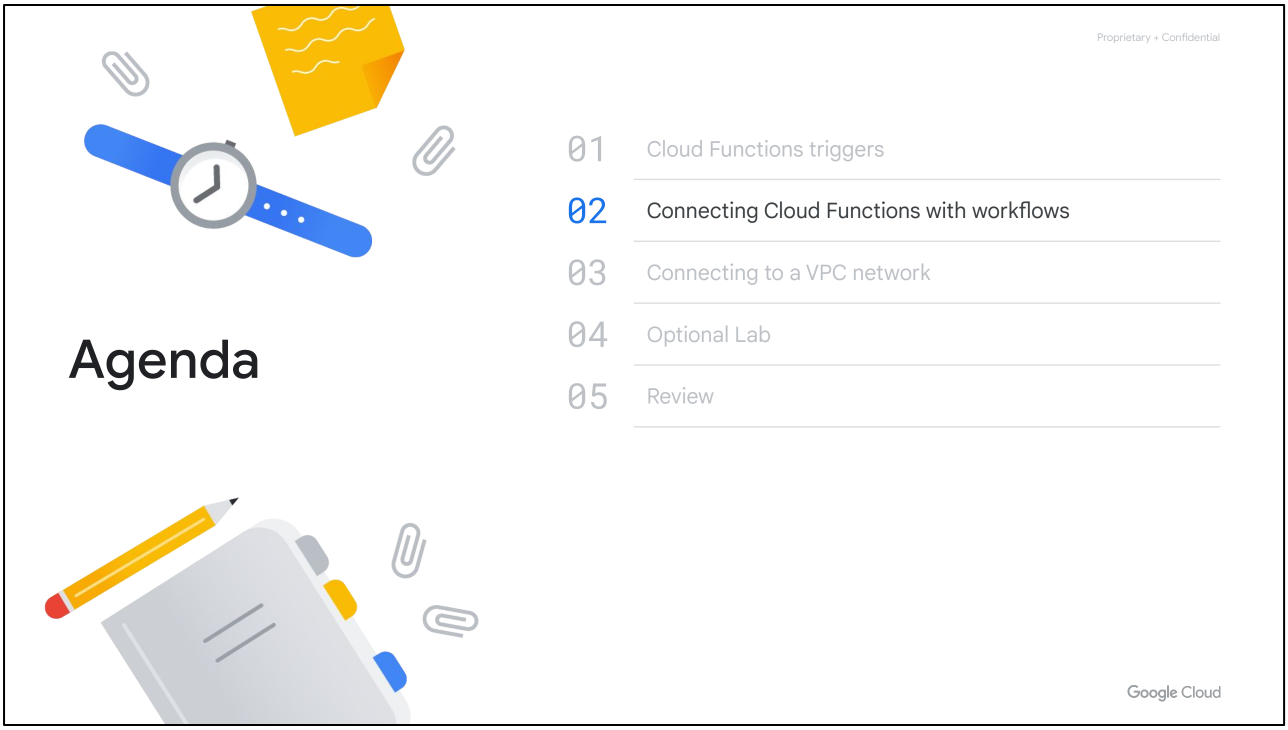
- Supported for various Firebase services:
  - Google Analytics for Firebase
  - Firebase Realtime Database
  - Firebase Authentication
  - Firebase Remote Config
- Triggers for each Firebase service use different event types and config resources.
- Handle events in the Firebase service that is in the same Google Cloud project as the function.

Google Cloud

Cloud Functions supports triggers for various Firebase services:

- Google Analytics for Firebase (1st gen only)
- Firebase Realtime Database
- Firebase Authentication (1st gen only)
- Firebase Remote Config

You can handle events in the Firebase service that is in the same Google Cloud project as the function.



Agenda

01 Cloud Functions triggers

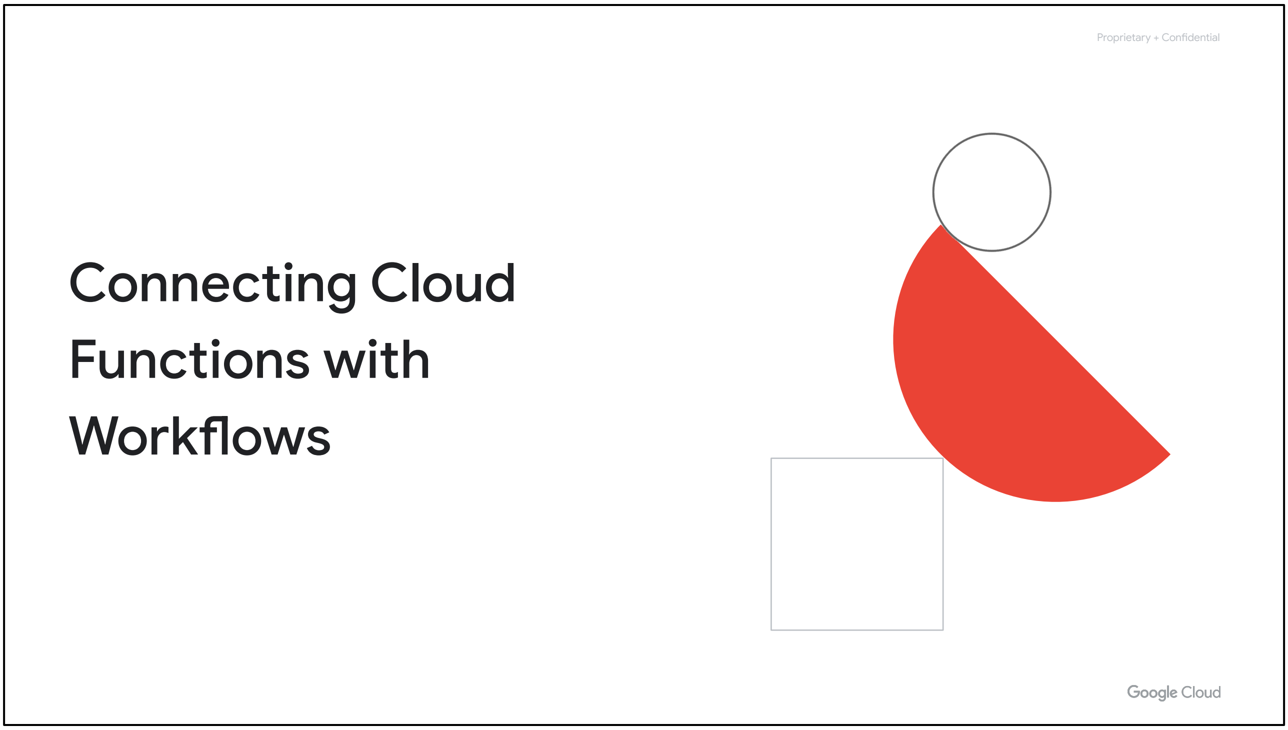
02 Connecting Cloud Functions with workflows

03 Connecting to a VPC network

04 Optional Lab

05 Review

Let’s now discuss how you can use Workflows to connect Cloud Functions and link a series of services together.

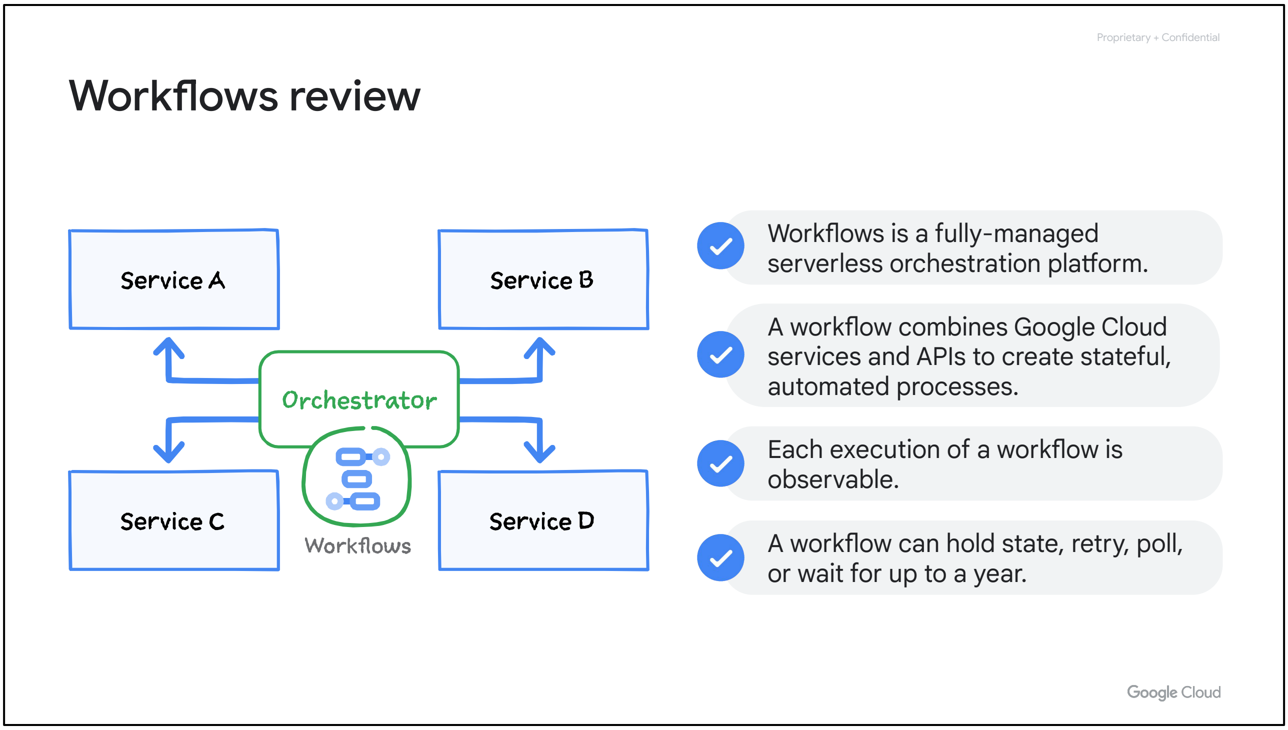


Proprietary + Confidential

Connecting Cloud Functions with Workflows

Google Cloud

What is Workflows?



Workflows review

- ✓ Workflows is a fully-managed serverless orchestration platform.
- ✓ A workflow combines Google Cloud services and APIs to create stateful, automated processes.
- ✓ Each execution of a workflow is observable.
- ✓ A workflow can hold state, retry, poll, or wait for up to a year.

Google Cloud

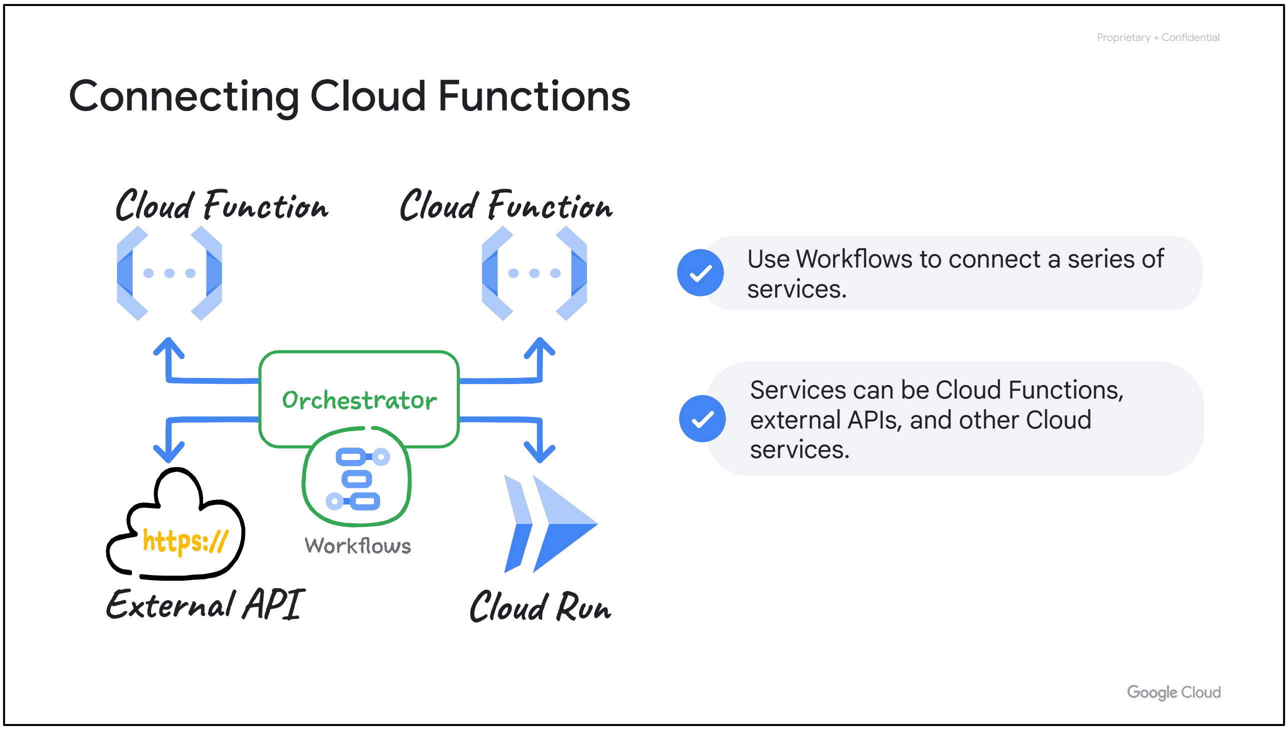
Workflows is a fully-managed, serverless orchestration platform that executes services in an order that you define (a workflow).

It acts as the central orchestrator for the service orchestration pattern. You design and deploy workflows, which orchestrate and combine Google Cloud services and API calls.

To build stateful, automated processes, Workflows can include custom services that are hosted on Cloud Run or Cloud Functions.

A workflow provides a central source-of-truth for the application flow. Each execution of a workflow is logged and is observable, which makes it easier to understand the current state of the workflow and troubleshoot any issues.

A workflow can hold state, retry, poll, or wait for up to a year. This flexibility allows for creation of long-running business processes.



Proprietary + Confidential

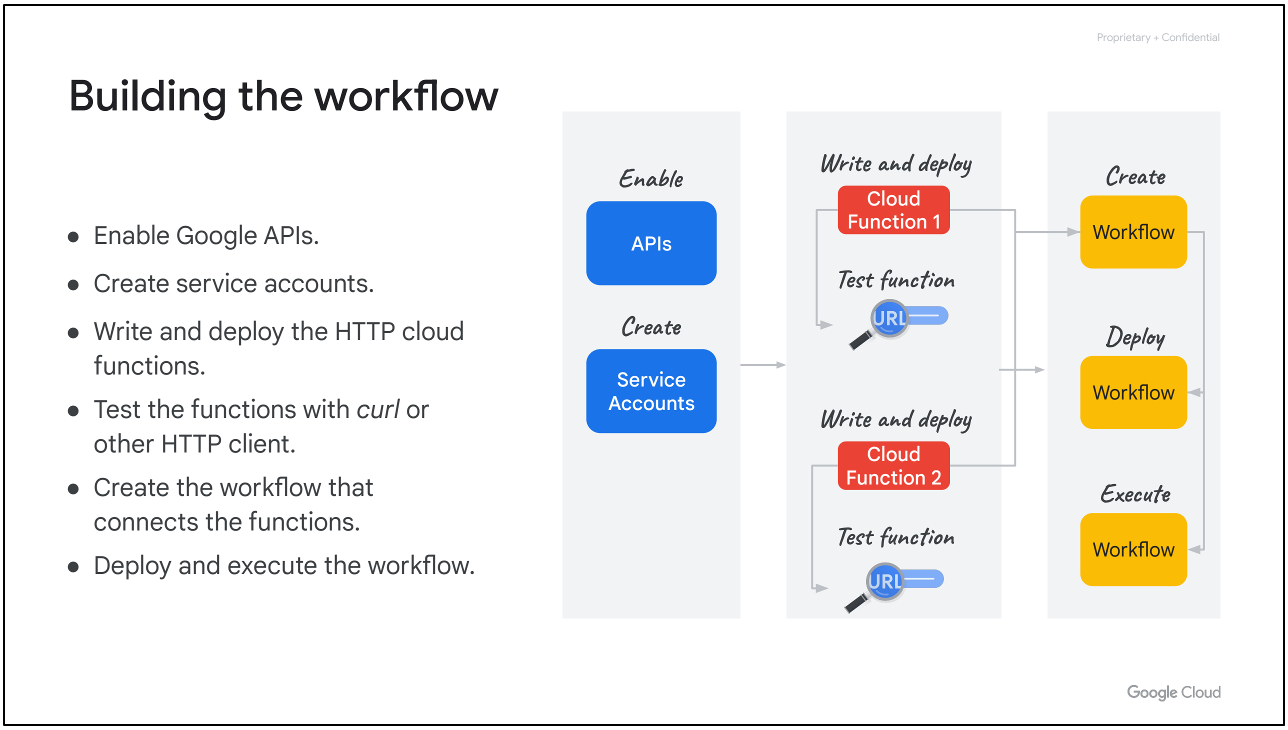
Connecting Cloud Functions

- ✓ Use Workflows to connect a series of services.
- ✓ Services can be Cloud Functions, external APIs, and other Cloud services.

Google Cloud

You can use Workflows to connect a series of services together that include HTTP services built with Cloud Functions, external APIs, and other Cloud services like Cloud Run.

With this approach, you can create a flexible serverless application.



Proprietary + Confidential

Building the workflow

- Enable Google APIs.
- Create service accounts.
- Write and deploy the HTTP cloud functions.
- Test the functions with *curl* or other HTTP client.
- Create the workflow that connects the functions.
- Deploy and execute the workflow.

Enable

APIs

Create

Service Accounts

Write and deploy

Cloud Function 1

Test function

URL

Write and deploy

Cloud Function 2

Test function

URL

Create

Workflow

Deploy

Workflow

Execute

Workflow

Google Cloud

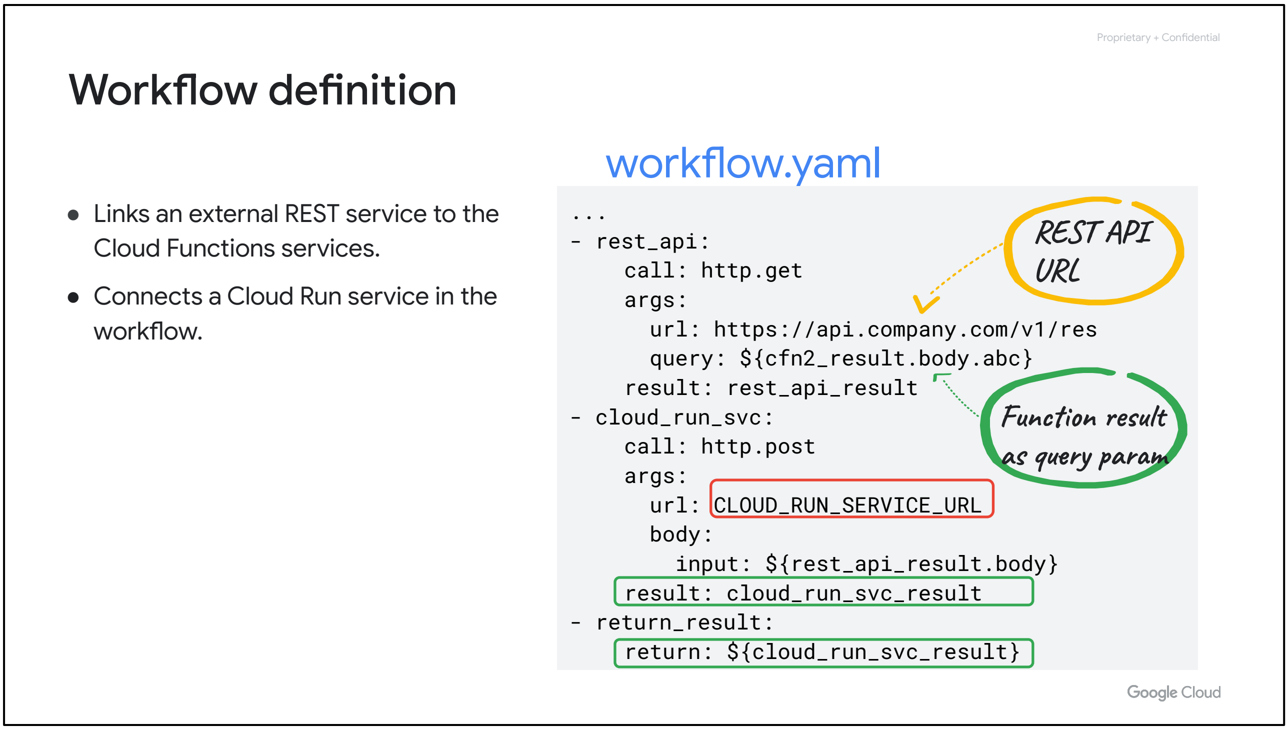
The first step to build a workflow is to enable the required Google APIs for Cloud Functions, Cloud Run, Workflows, and any other services that you use.

You may also need to create any service accounts that are required to access these services.

Next, write, and deploy the functions. These functions are HTTP functions with HTTP triggers that generate URL endpoints that are used to invoke the functions.

Test the functions individually with *curl* or any other HTTP client. It's also a best practice to test the functions locally before deployment.

You then create the workflow that connects the Cloud Functions. After the workflow is created, you deploy and execute it.



Proprietary + Confidential

Workflow definition

- Links an external REST service to the Cloud Functions services.
- Connects a Cloud Run service in the workflow.

workflow.yaml

```
...
- rest_api:
    call: http.get
    args:
      url: https://api.company.com/v1/res
      query: ${cfn2_result.body.abc}
    result: rest_api_result
- cloud_run_svc:
    call: http.post
    args:
      url: CLOUD_RUN_SERVICE_URL
      body:
        input: ${rest_api_result.body}
    result: cloud_run_svc_result
- return_result:
    return: ${cloud_run_svc_result}
```

REST API URL

Function result as query param

Google Cloud

A workflow is made up of a series of steps that are described using the Workflows syntax. The set of steps is the workflow definition and can be written in either YAML or JSON format.

In the sample workflow definition:

- The Cloud Functions steps cfn1 and cfn2, are invoked from the workflow through an HTTP request using the GET and POST methods respectively.
- The URLs to the functions are provided as arguments in the function definition.
- The result generated by the first cloud function is provided as input to the second cloud function.

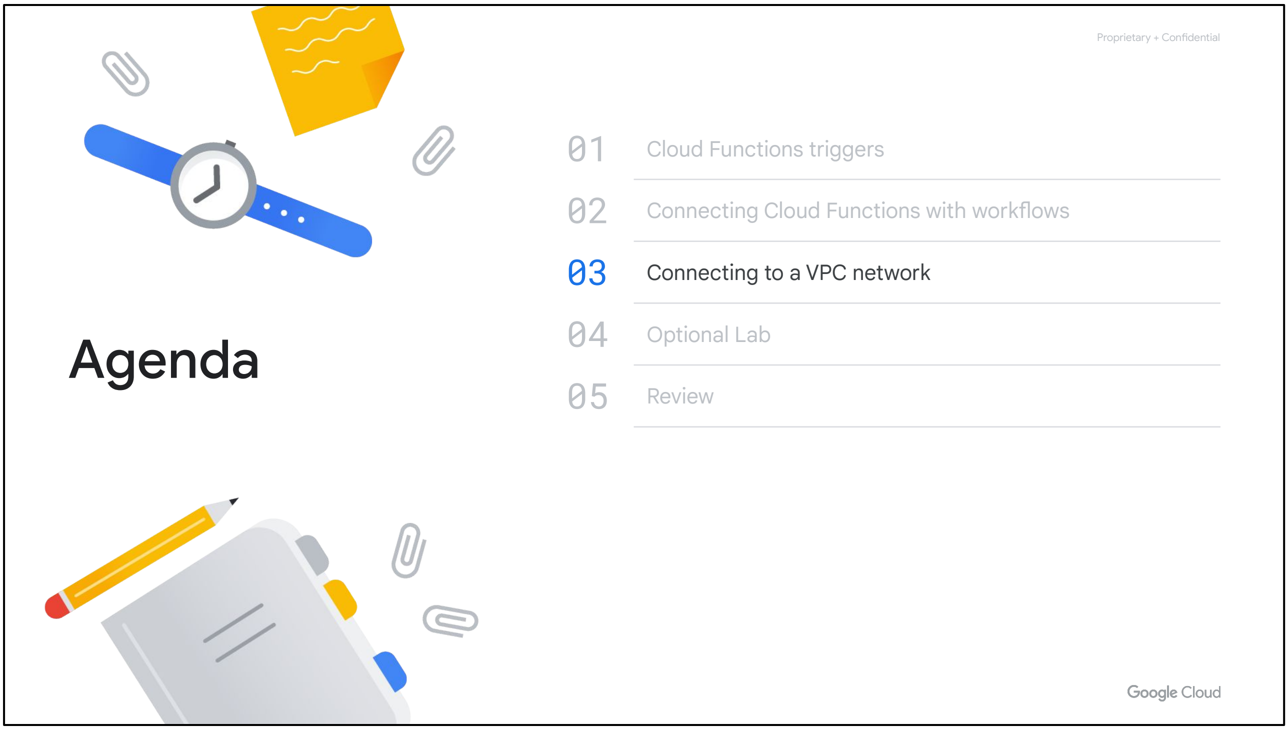
The workflow definition also includes:

- The configuration to connect to an external REST API endpoint with the result of the 2nd cloud function passed in a query parameter.

and,

- Configuration that connects a Cloud Run service in the workflow.

The result generated by the Cloud Run service is the result of the workflow.



Agenda

01 Cloud Functions triggers

02 Connecting Cloud Functions with workflows

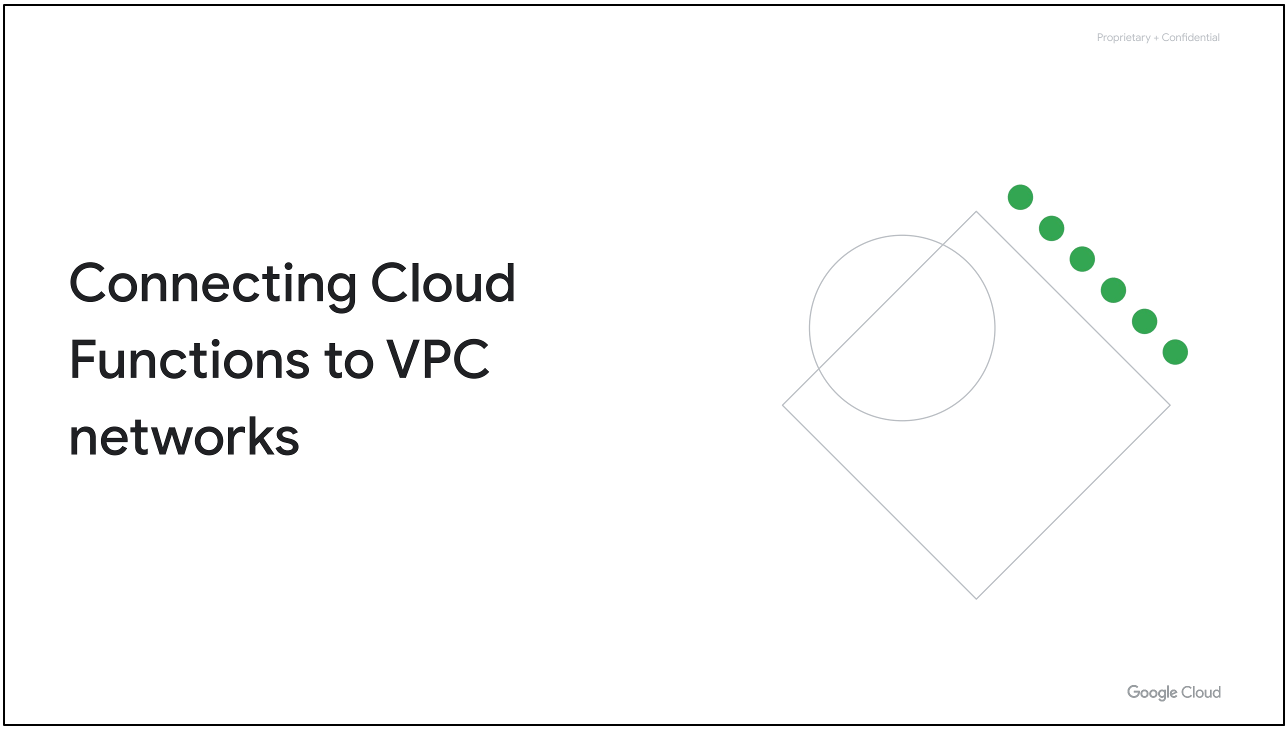
03 Connecting to a VPC network

04 Optional Lab

05 Review

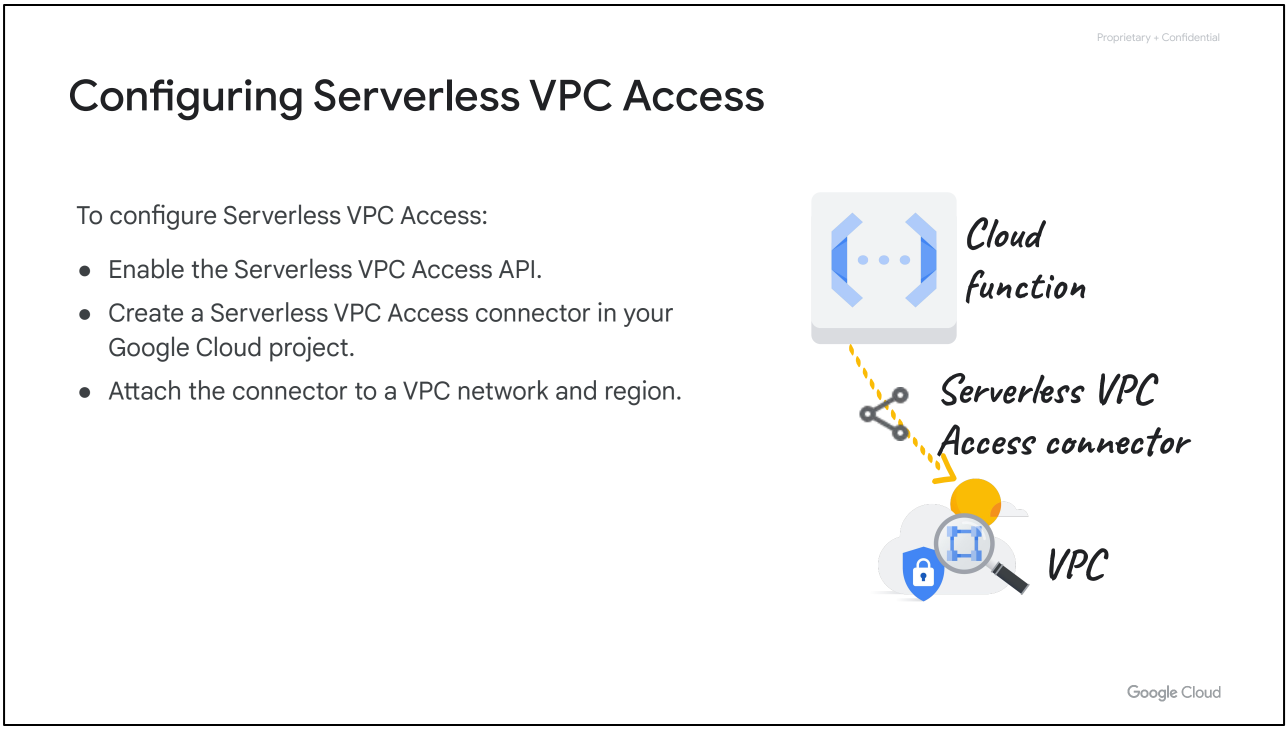
Google Cloud

In this section, we discuss how you can connect Cloud Functions to resources in a VPC network.



Connecting Cloud Functions to VPC networks

You use Serverless VPC Access to connect Cloud Functions directly to your VPC network.



Configuring Serverless VPC Access

To configure Serverless VPC Access:

- Enable the Serverless VPC Access API.
- Create a Serverless VPC Access connector in your Google Cloud project.
- Attach the connector to a VPC network and region.

Cloud function
Serverless VPC Access connector
VPC

Google Cloud

A Virtual Private Cloud (VPC) network is a virtual version of a physical network, implemented inside Google's production network.

It’s a global resource that consists of a list of regional virtual subnetworks (subnets) in data centers, all connected by a global wide area network.

With Serverless VPC Access, you can connect Cloud Functions directly to your VPC network, and enable access to Compute Engine VM instances, Memorystore, and other resources with an internal IP address.

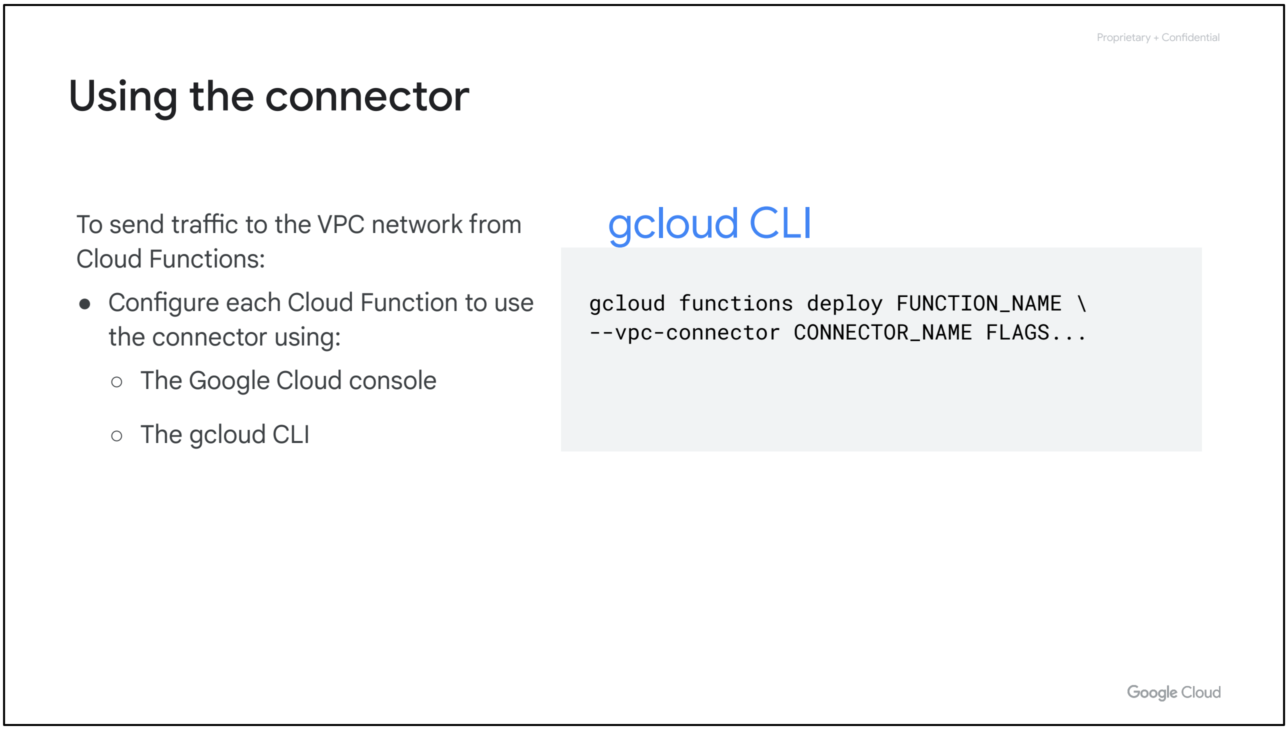
With Serverless VPC Access, you can send requests and receive responses to and from your VPC network using internal DNS and internal IP addresses, so that traffic is not exposed to the internet.

To configure Serverless VPC Access, you:

- Enable the Serverless VPC Access API.
- Create a Serverless VPC Access connector in your Google Cloud project.
- Attach the connector to a VPC network and region.

A Serverless VPC Access connector is a resource that handles traffic between your serverless Cloud Functions environment and your VPC network.
The region that is configured for the connector must match the region where your Cloud Functions is deployed.

A subnet or CIDR range must be configured exclusively for use by the connector.



Using the connector

To send traffic to the VPC network from Cloud Functions:

- Configure each Cloud Function to use the connector using:
  - The Google Cloud console
  - The gcloud CLI

gcloud CLI

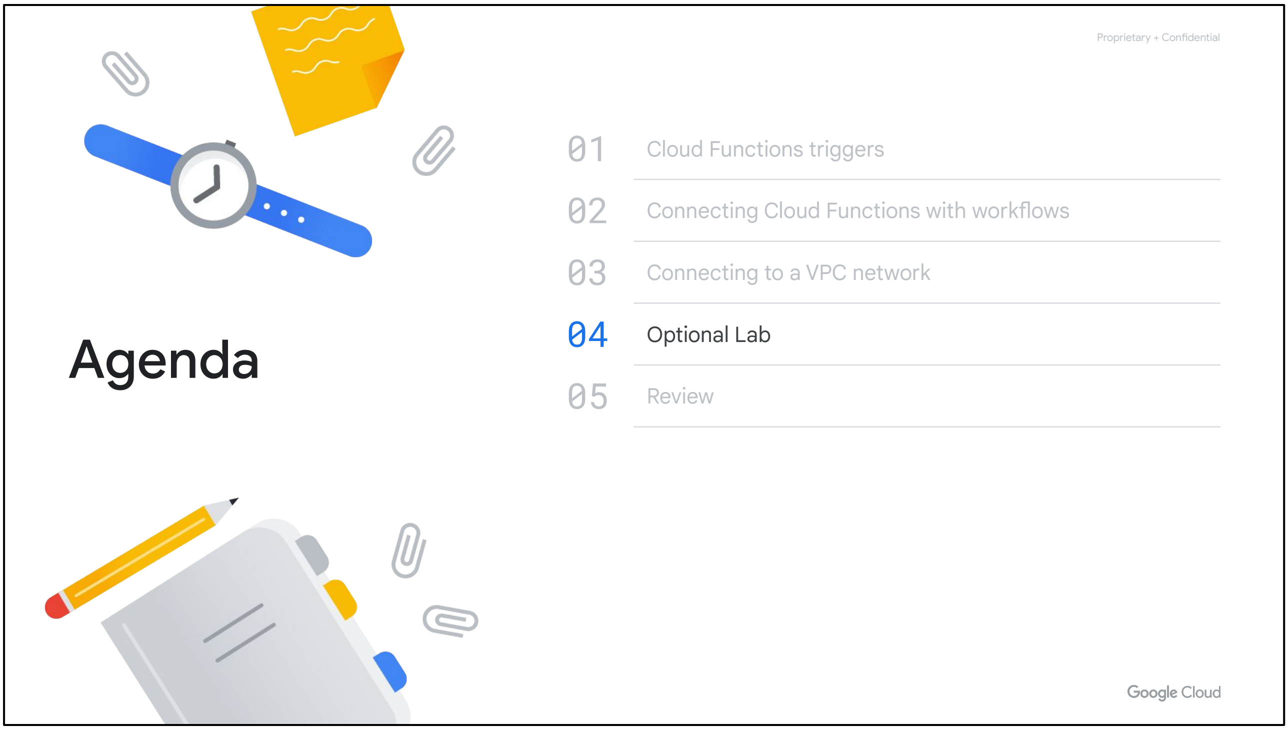
```
gcloud functions deploy FUNCTION_NAME \
--vpc-connector CONNECTOR_NAME FLAGS...
```

After you have created a Serverless VPC Access connector, you must configure each function that you want to connect to your VPC network.

You can configure a function to use a connector from the Google Cloud console or the gcloud CLI.

You can also restrict your connector's access to resources in your VPC network by using firewall rules, and connect Cloud Functions to resources in a shared VPC network.

For more information on configuring Serverless VPC Access connectors, refer to the reading document provided with this course, and the Cloud Functions website documentation.



Agenda

01 Cloud Functions triggers

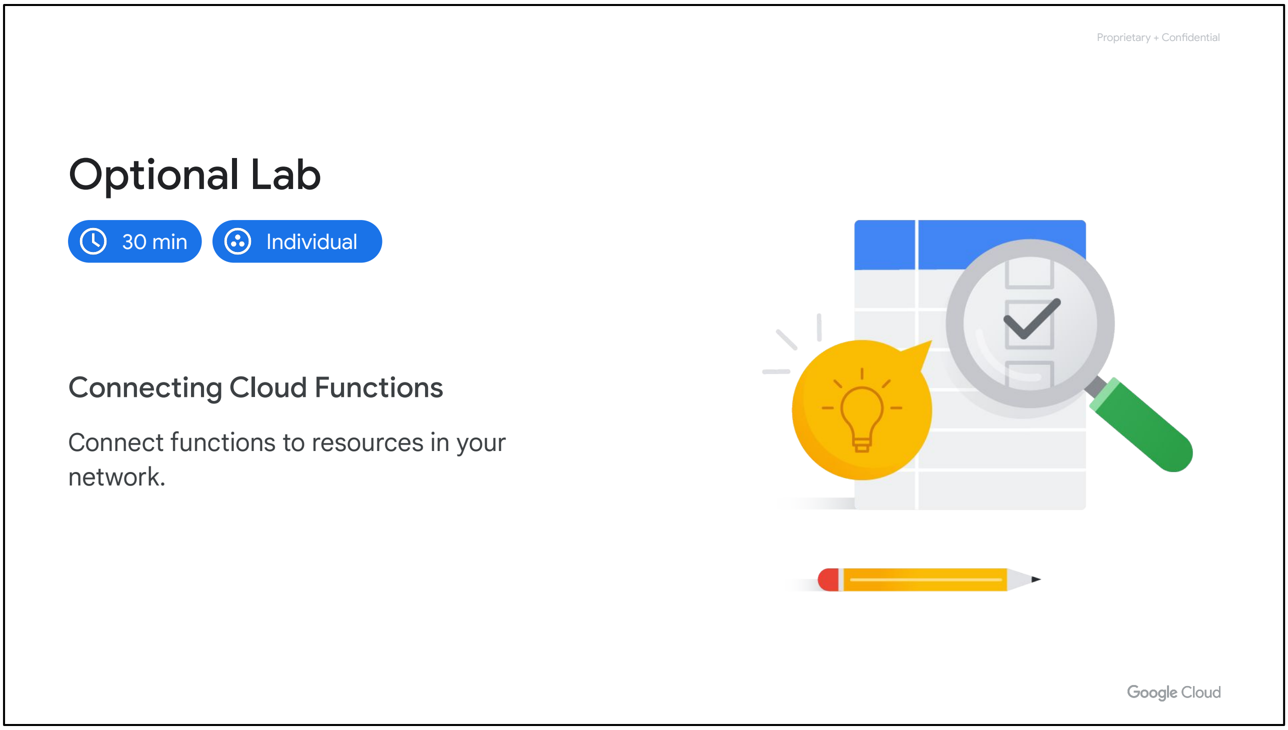
02 Connecting Cloud Functions with workflows

03 Connecting to a VPC network

04 Optional Lab

05 Review

This is an optional lab that teaches you how to connect Cloud Functions to resources in a VPC network.



Proprietary + Confidential

Optional Lab

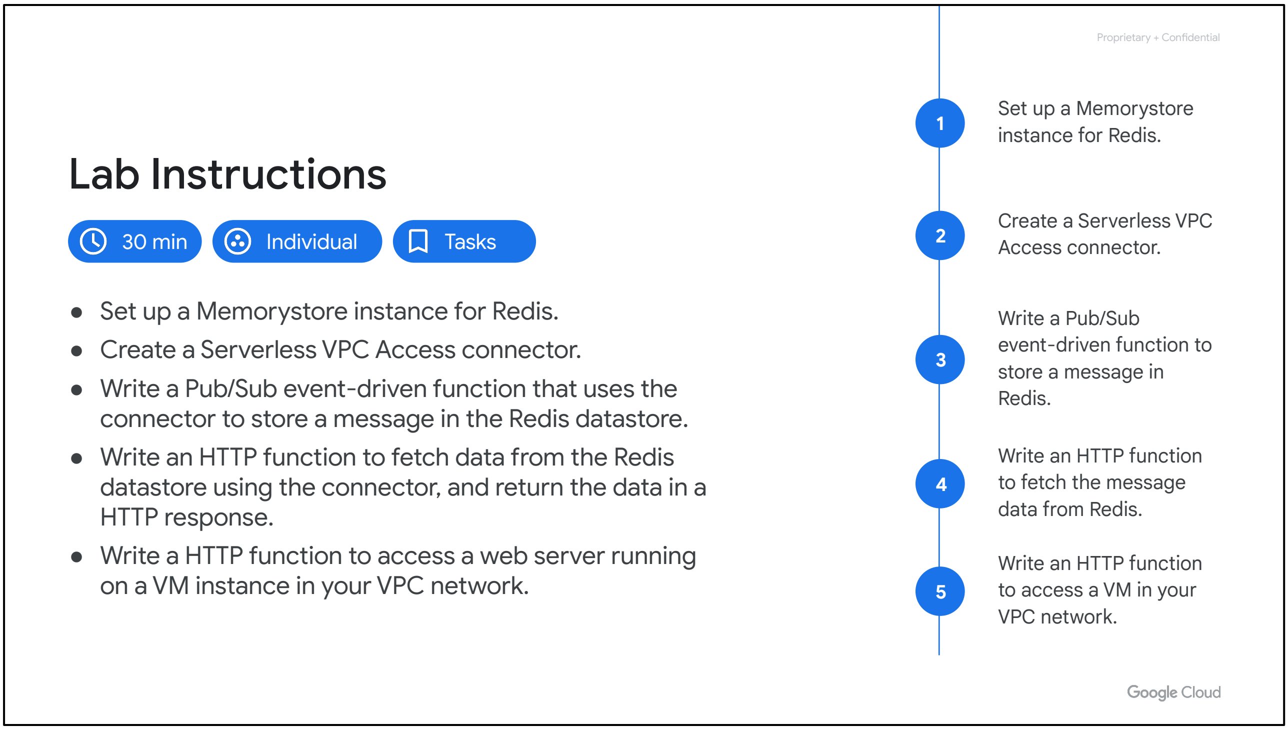
30 min | Individual

Connecting Cloud Functions

Connect functions to resources in your network.

Google Cloud

In this lab, you create a Serverless VPC Access connector to connect Cloud Functions to a Memorystore for Redis instance, and to a VM instance in your VPC network.



Lab Instructions

30 min | Individual | Tasks

- Set up a Memorystore instance for Redis.
- Create a Serverless VPC Access connector.
- Write a Pub/Sub event-driven function that uses the connector to store a message in the Redis datastore.
- Write an HTTP function to fetch data from the Redis datastore using the connector, and return the data in a HTTP response.
- Write a HTTP function to access a web server running on a VM instance in your VPC network.

1 Set up a Memorystore instance for Redis.

2 Create a Serverless VPC Access connector.

3 Write a Pub/Sub event-driven function to store a message in Redis.

4 Write an HTTP function to fetch the message data from Redis.

5 Write an HTTP function to access a VM in your VPC network.

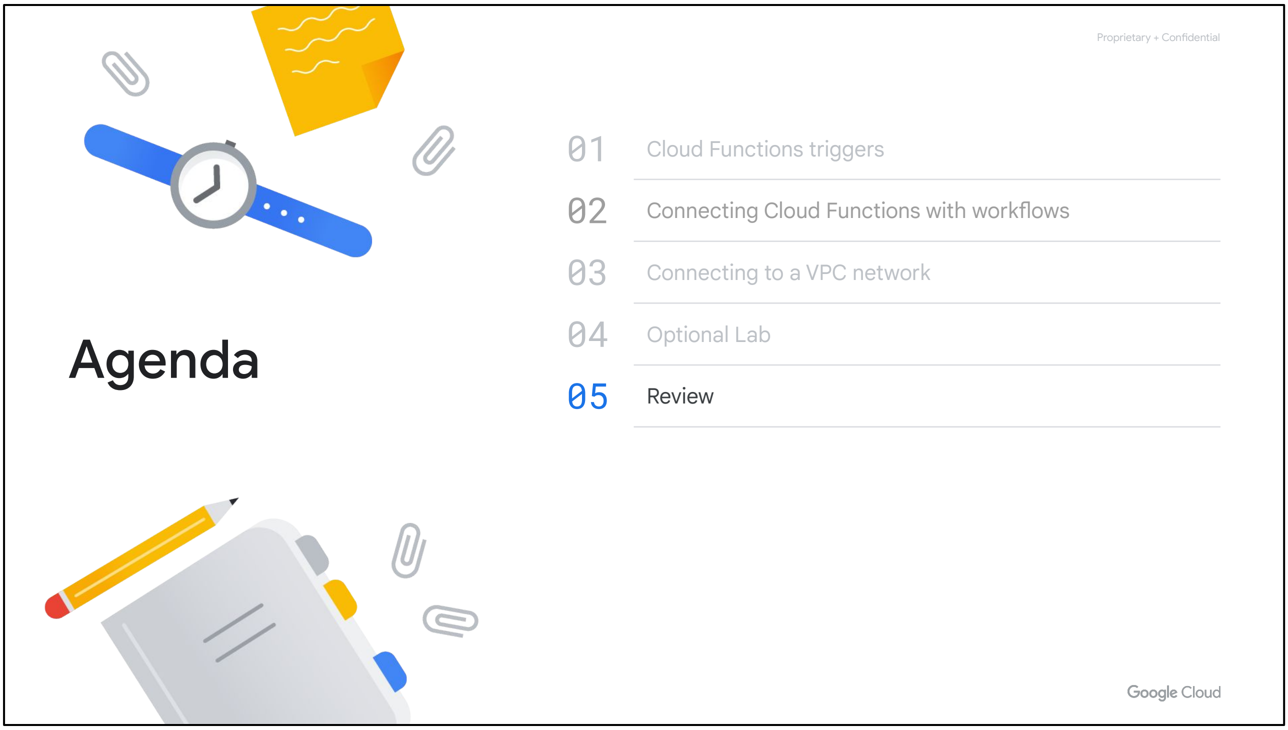
Memorystore is a fully managed in-memory data store service for Redis and Memcached on Google Cloud. You first create a Memorystore instance for Redis.

Next, you create a Serverless VPC Access connector. A connector enables your Cloud Functions to connect to resources in a VPC network.

Then, you write and deploy an event-driven function that receives a message from a Pub/Sub topic and, using the connector, stores the message data in the Memorystore for Redis datastore.

You also write a HTTP function to fetch the message data from the Redis datastore using the connector, and return the data in a HTTP response.

You then create a VM with a simple web server in your network, and write a HTTP function to access the web server using the Serverless VPC Access connector.



Agenda

01 Cloud Functions triggers

02 Connecting Cloud Functions with workflows

03 Connecting to a VPC network

04 Optional Lab

05 Review

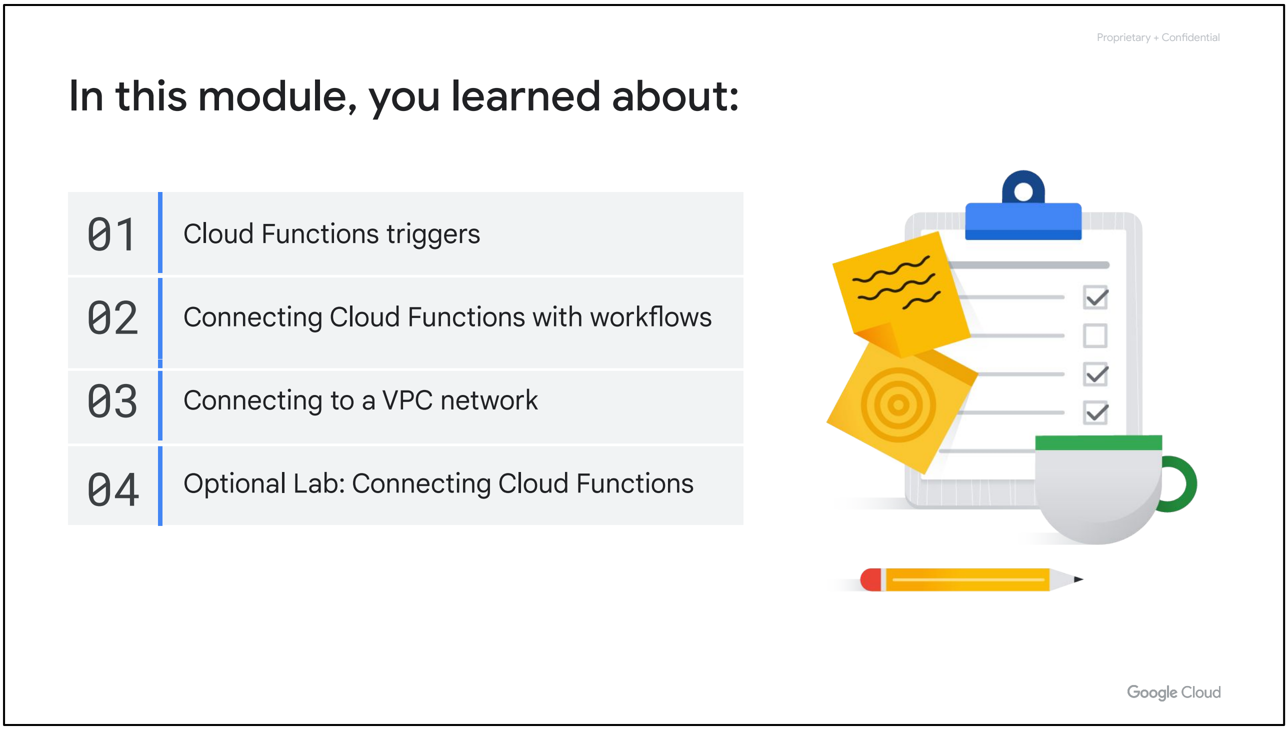
Google Cloud

Let’s review what was discussed in this module.

Google Cloud

Review: Calling and Connecting Cloud Functions

In this module, we discussed triggers and how they are used to trigger Cloud Functions. We also discussed how to connect functions with Workflows, and how to connect functions to resources in a VPC network.



Proprietary + Confidential

In this module, you learned about:

01 Cloud Functions triggers

02 Connecting Cloud Functions with workflows

03 Connecting to a VPC network

04 Optional Lab: Connecting Cloud Functions

Google Cloud

In this module, you learned about HTTP and event triggers that are used to invoke Cloud Functions.

You learned how to connect Cloud Functions with Workflows, Google Cloud's serverless orchestration platform that executes cloud services as a series of steps in a workflow.

We discussed how you can connect Cloud Functions to internal resources in a VPC network, and reviewed an optional lab to connect Cloud Functions to Memorystore and a VM resource in a network.