Google Cloud

Blog

Google Cloud

Blog

Google Cloud

Blog

Developers & Practitioners

Eventarc: A unified eventing experience in Google Cloud

January 16, 2021

Google Cloud

Blog

Mete Atamel
Developer Advocate

I recently talked about orchestration versus choreography in connecting microservices and introduced Workflows for use cases that can benefit from a central orchestrator. I also mentioned Eventarc and Pub/Sub in the choreography camp for more loosely coupled event-driven architectures.

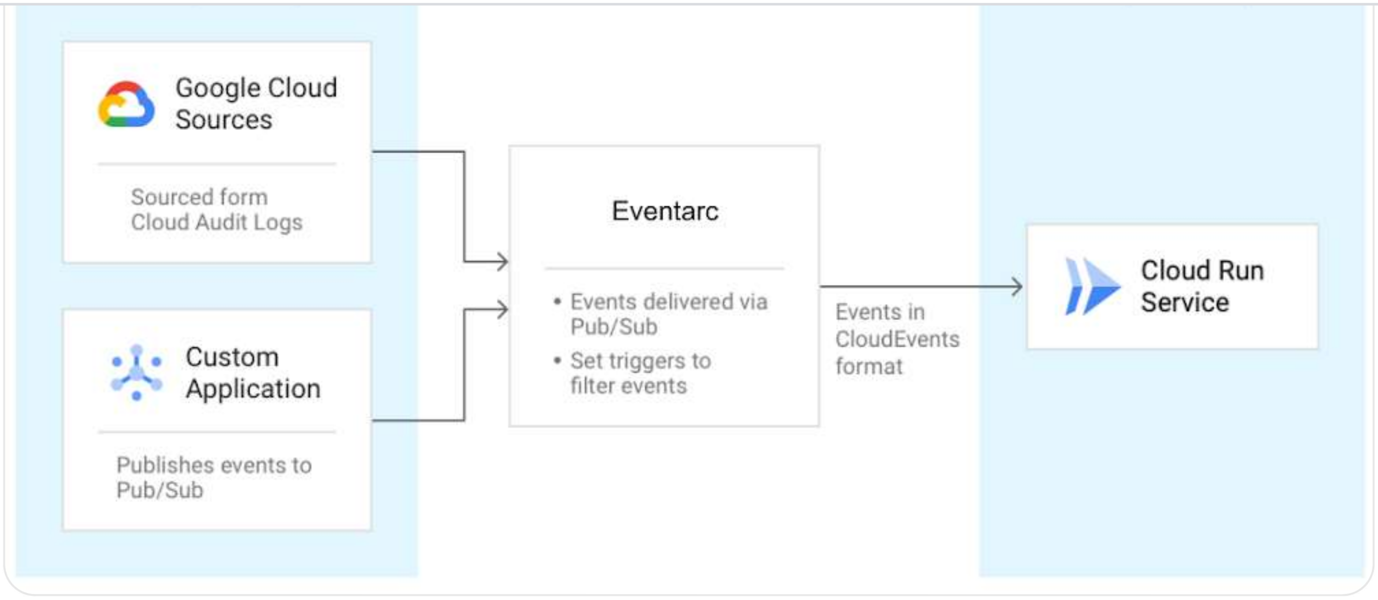
In this blog post, I talk more about the unified eventing experience by Eventarc.

What is Eventarc?

We announced Eventarc back in October as a new eventing functionality that enables you to send events to Cloud Run from more than 60 Google Cloud sources. It works by reading Audit Logs from various sources and sending them to Cloud Run

Google Cloud

Blog


Google Cloud Sources
Sourced form Cloud Audit Logs
Custom Application
Publishes events to Pub/Sub
Eventarc
• Events delivered via Pub/Sub
• Set triggers to filter events
Events in CloudEvents format
Cloud Run Service

Getting events to Cloud Run

There are already other ways to get events to Cloud Run, so you might wonder what's special about Eventarc? I'll get to this question, but let's first explore one of those ways, Pub/Sub.

As shown in this Using Pub/Sub with Cloud Run tutorial, Cloud Run services can receive messages pushed from a Pub/Sub topic. This works if the event source can directly publish messages to a Pub/Sub topic. It can also work for services that have integration with Pub/Sub and publish their events through that integration. For example, Cloud Storage is one of those services and in this tutorial, I show how to receive updates from a Cloud Storage bucket using a Pub/Sub topic in the middle.

For other services with no integration to Pub/Sub, you have to either integrate them with Pub/Sub and configure Pub/Sub to route messages to Cloud Run or you need to find another way of sourcing those events. It's possible but definitely not trivial. That's where Eventarc comes into play.

Any service with Audit Log integration or any application that can send a message to a Pub/Sub topic can be event sources for Eventarc. You don't have to worry about the underlying infrastructure with Eventarc. It is a managed service with no clusters to set up or maintain.

It also has some concrete benefits beyond the easy integration. It provides consistency and structure to how events are generated, routed, and consumed. Let's explore those benefits next.

Simplified and centralized routing

Eventarc introduces the notion of a trigger. A trigger specifies routing rules from event sources to event sinks. For example, one can listen for new object creation events in Cloud Storage and route them to a Cloud Run service by simply creating an Audit Log trigger as follows:

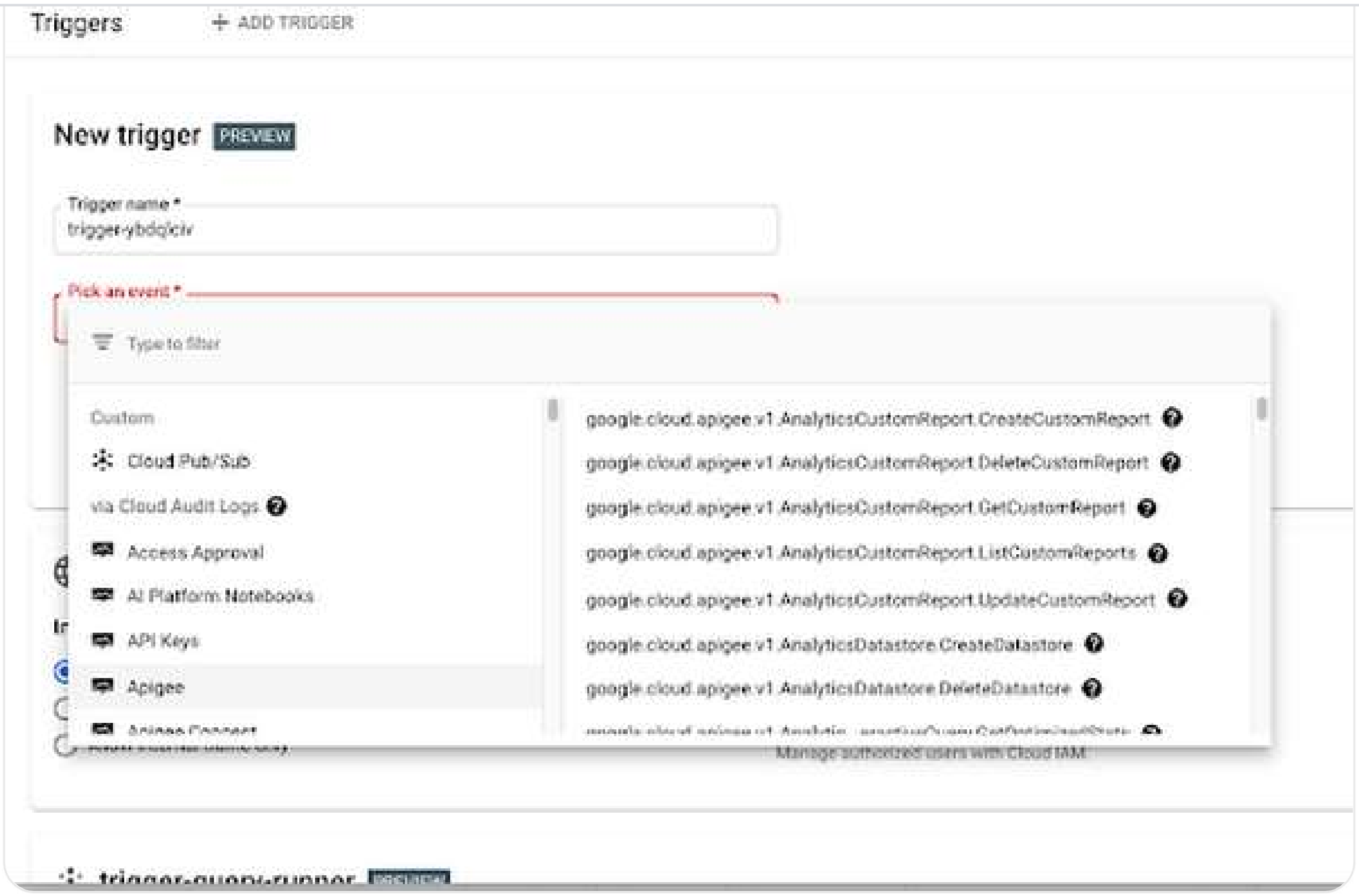
```
gcloud beta eventarc triggers create trigger-auditlog \
  --destination-run-service=${SERVICE_NAME} \
  --destination-run-region=${REGION}
  --matching-criteria="type=google.cloud.audit.log.v1.written" \
  --matching-criteria="serviceName=storage.googleapis.com" \
  --matching-criteria="methodName=storage.objects.create" \
  --service-account=${PROJECT_NO}-
compute@developer.gserviceaccount.com
```

If you want to listen for messages from Pub/Sub instead, that's another trigger:

```
criteria="type=google.cloud.pubsub.topic.v1.messagePublished"
```

This trigger creates a Pub/Sub topic under the covers. Applications can send messages to that topic and those messages are routed to the specified Cloud Run service by Eventarc.

Users can also create triggers from Google Cloud Console under the triggers section of Cloud Run:



trigger-ybdqlciv

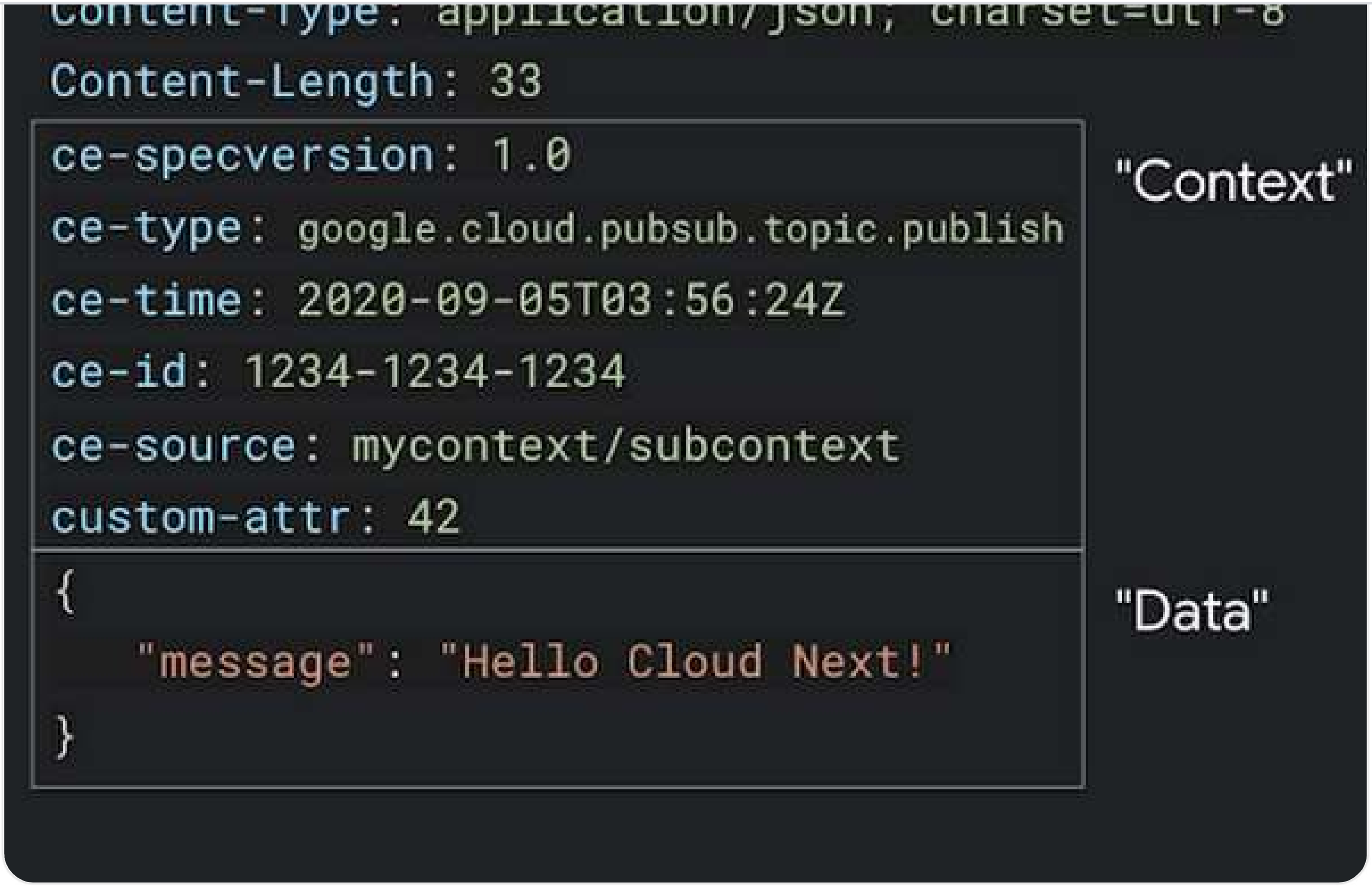
google.cloud.apigee.v1.AnalyticsCustomReport.CreateCustomReport
google.cloud.apigee.v1.AnalyticsCustomReport.DeleteCustomReport
google.cloud.apigee.v1.AnalyticsCustomReport.GetCustomReport
google.cloud.apigee.v1.AnalyticsCustomReport.ListCustomReports
google.cloud.apigee.v1.AnalyticsCustomReport.UpdateCustomReport
google.cloud.apigee.v1.AnalyticsDatastore.CreateDatastore
google.cloud.apigee.v1.AnalyticsDatastore.DeleteDatastore

By having event routing defined as triggers, users can list and manage all their triggers in one central place in Eventarc. Here's the command to see all created triggers:

```
gcloud beta eventarc triggers list
```

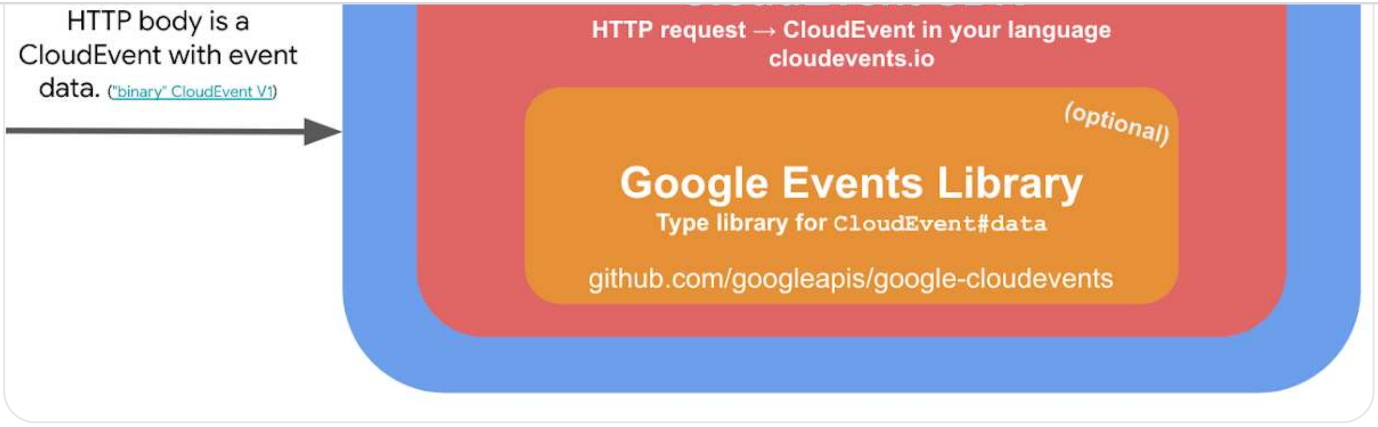
Consistency with eventing format and libraries

In Eventarc, different events from different sources are converted to CloudEvents compliant events. CloudEvents is a specification for describing event data in a common way with the goal of consistency, accessibility and portability.



```
Content-Type: application/json; charset=utf-8
Content-Length: 33
ce-specversion: 1.0
ce-type: google.cloud.pubsub.topic.publish
ce-time: 2020-09-05T03:56:24Z
ce-id: 1234-1234-1234
ce-source: mycontext/subcontext
custom-attr: 42
{
    "message": "Hello Cloud Next!"
}
```

Event consumers can read these events directly. We also try to make it easier in various languages (Node.js, Python, Go, Java, C# and more) with CloudEvents SDKs to read the event and Google Events libraries to parse the date field.


HTTP body is a CloudEvent with event data. ("binary" CloudEvent V1)
HTTP request → CloudEvent in your language
cloudevents.io
(optional)
Google Events Library
Type library for CloudEvent#data
github.com/googleapis/google-cloudevents

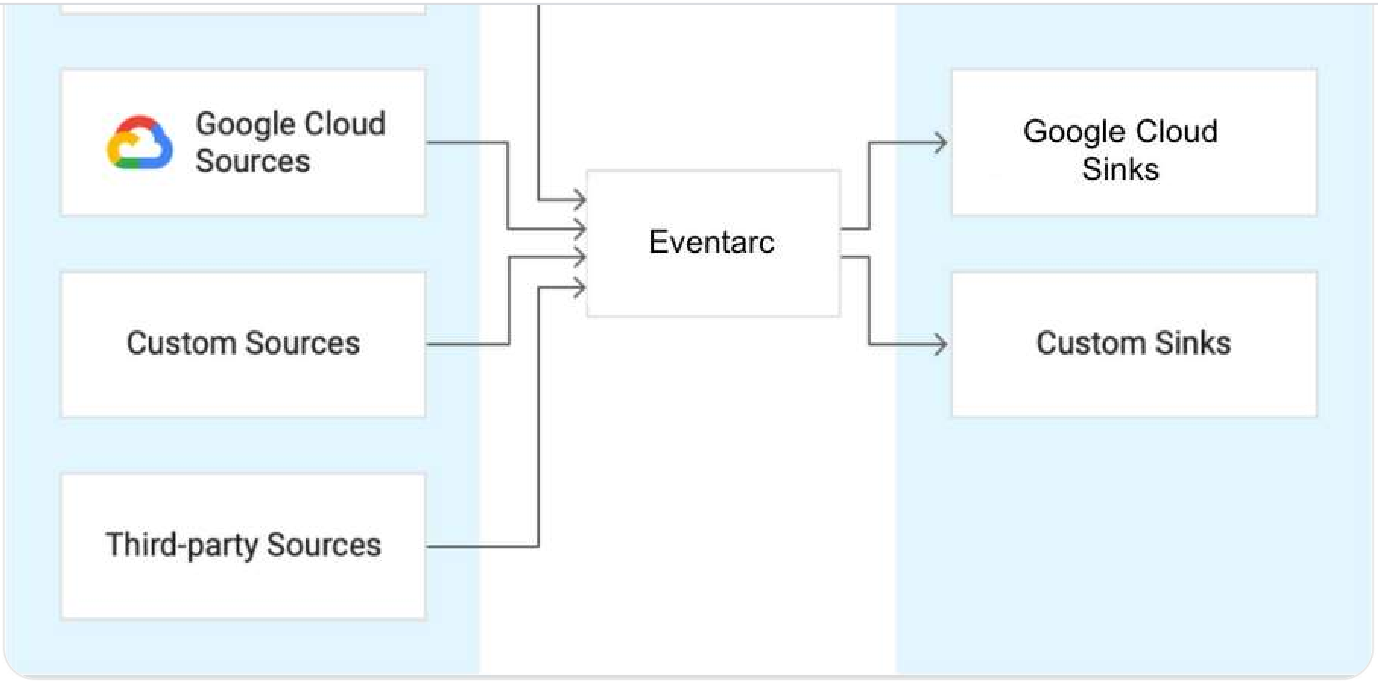
Going back to our Cloud Storage example earlier, this is how you'd read Cloud Storage events via AuditLogs in Node.js using the two mentioned libraries:

```
const { HTTP } = require("cloudevents");
const {toLogEntryData} =
require('@google/events/cloud/audit/v1/LogEntryData')

app.post('/', async (req, res) => {

  // Read CloudEvent using CloudEvents SDK
  const cloudEvent = HTTP.toEvent({ headers: req.headers, body:
req.body });

 // Read AuditLog using Google.Events library for Node.js
 const logEntryData = toLogEntryData(cloudEvent.data);

 // Extract bucket and objectName
 const tokens = logEntryData.protoPayload.resourceName.split('/');
 const bucket = tokens[3];
```

```
using CloudNative.CloudEvents;
using Google.Events;
using Google.Events.Protobuf.Cloud.PubSub.V1;

public async Task<CloudEvent> Read(HttpContext context) {
  // Read CloudEvent using CloudEvents SDK
  var cloudEvent = await context.Request.ReadCloudEventAsync();

 // Read Pub/Sub message using Google.Events library for .NET
 var messagePublishedData =
CloudEventConverters.ConvertCloudEventData<MessagePublishedData>
(cloudEvent);

 // Extract the Pub/Sub message
 var pubSubMessage = messagePublishedData.Message;
```

Long term vision

The long term vision of Eventarc is to be the hub of events from more sources and sinks, enabling a unified eventing story in Google Cloud and beyond.



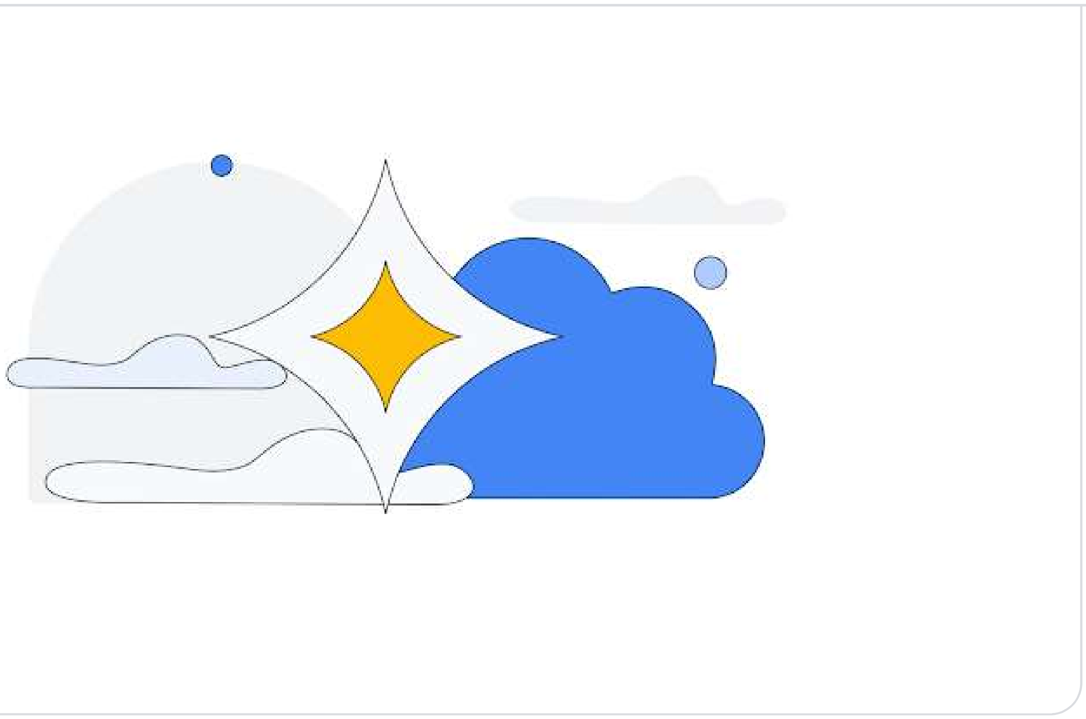
In the future, you can expect to read events directly (without having to go through Audit Logs) from more Google Cloud sources (eg. Firestore, BigQuery, Storage), Google sources (eg. Gmail, Hangouts, Chat), 3rd party sources (eg. Datadog, PagerDuty) and send these events to more Google Cloud sinks (eg. Cloud Functions, Compute Engine, Pub/Sub) and custom sinks (any HTTP target).

Now that you have a better overall picture of the current state and future vision for Eventarc:

- Check out the Eventarc documentation.
- Check out Trigger Cloud Run with events from Eventarc for a hands-on codelab.
- Send us feedback on Eventarc and which sources and sinks you would value the most.

Google Cloud

Blog



Serverless

Trigger Cloud Run with events from more than 60 Google Cloud sources

Now, you can invoke applications running on Cloud Run with events generated by over 60 Google Cloud services.

By Prashant Gulati • 3-minute read

Posted in Developers & Practitioners—Google Cloud—Serverless

Related articles

AI & Machine Learning

Supercharging Vertex AI with Colab Enterprise and MLOps for generative AI

Google Cloud

Blog

models, tooling

By Amin Vahdat • 7-minute read

Developers & Practitioners

Google Cloud Innovators and Community join forces to boost your success

By Crystal Smith • 2-minute read

Cost Management

Keep a closer eye on Google Cloud costs with new Budgets for project users

By Mark Mirchandani • 3-minute read

Follow us

Google Cloud Google Cloud Products Privacy Terms

Help English