www.cloudskillsboost.google /focuses/15657

Eventarc for Cloud Run | Google Cloud Skills Boost

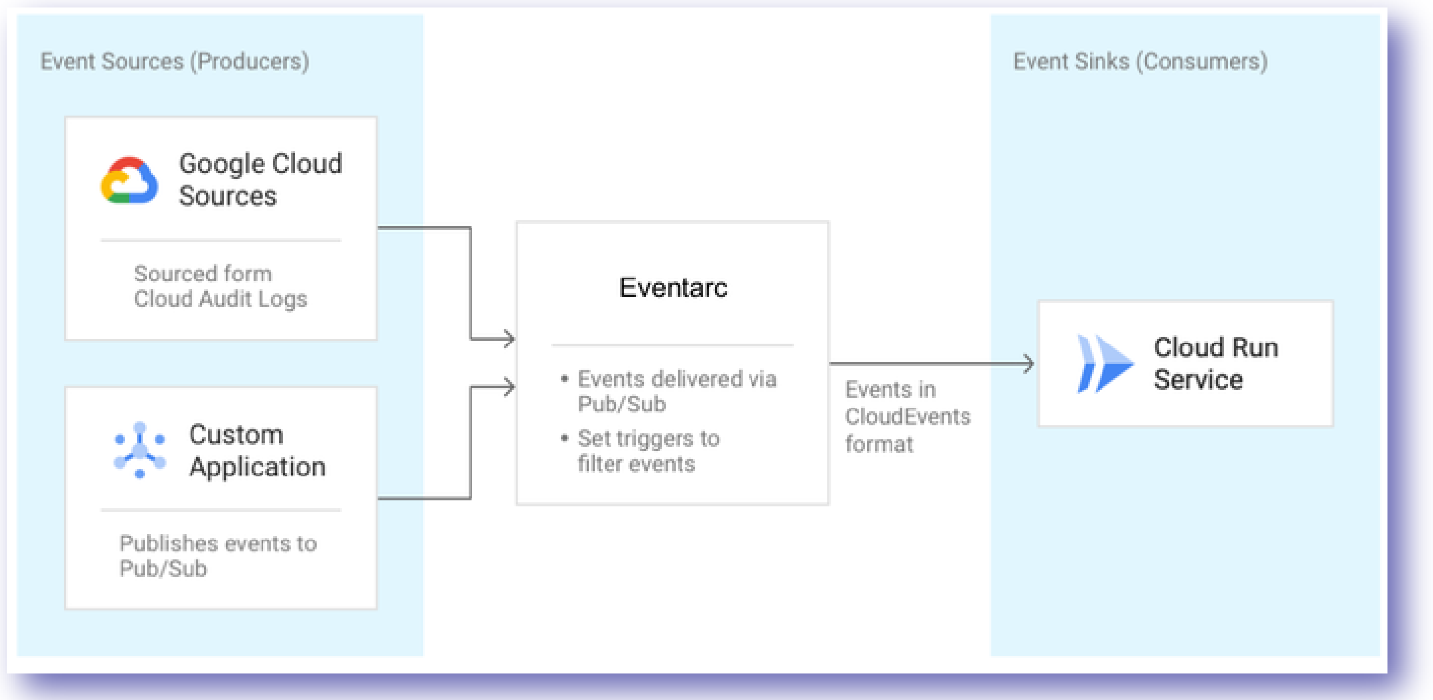
Qwiklabs : 14-17 minutes

GSP773

Google Cloud Self-Paced Labs

Overview

In this lab, you will learn how to utilize events for Cloud Run to manage communication between producer and consumers. Producers (i.e. Event Sources) provide the originating data. The data produced is sent to a Consumer (i.e. Event Sinks) that use the information passed. The diagram below provides a high level overview of this approach on Google Cloud:


Event Sources (Producers)
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
Event Sinks (Consumers)
Cloud Run Service

The unifying delivery mechanism between producers and consumers is Eventarc for Cloud Run. In the above example, Cloud Pub/Sub facilitates event delivery for their project events generated.

At the end of this lab, you will be able to deliver events from various sources to Google Cloud sinks and Custom sinks.

What you'll learn:

- Eventarc for Cloud Run
- Create a Cloud Run sink

- Create an Event trigger for Cloud Pub/Sub
- Create an Event trigger for Audit Logs

Prerequisites

Based on the content, it is recommended to have some familiarity with:

- Cloud Run
- Cloud Pub/Sub
- Logging

It is recommended to run this lab in an Incognito browser window.

Setup and requirements

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

Note: Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

- Time to complete the lab---remember, once you start, you cannot pause a lab.

Note: If you already have your own personal Google Cloud account or project, do not use it for this lab to avoid extra charges to your account.

How to start your lab and sign in to the Google Cloud Console

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is the **Lab Details** panel with the following:
   - The **Open Google Console** button
   - Time remaining
   - The temporary credentials that you must use for this lab
   - Other information, if needed, to step through this lab
2. Click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.

Tip: Arrange the tabs in separate windows, side-by-side.

Note: If you see the **Choose an account** dialog, click **Use Another Account**.

3. If necessary, copy the **Username** from the **Lab Details** panel and paste it into the **Sign in** dialog. Click **Next**.

4. Copy the **Password** from the **Lab Details** panel and paste it into the **Welcome** dialog. Click **Next**.

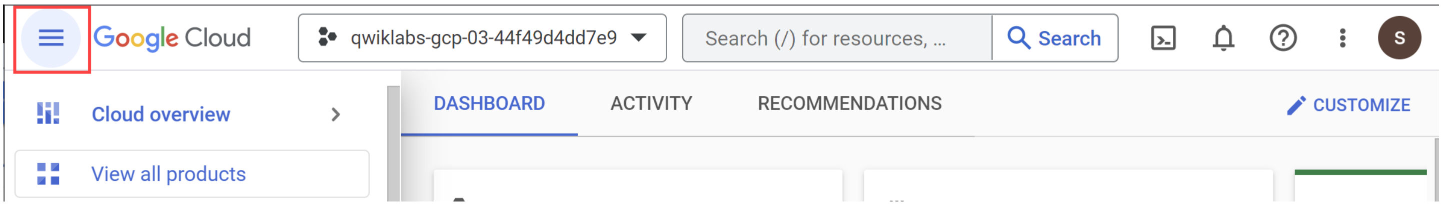
   Important: You must use the credentials from the left panel. Do not use your Google Cloud Skills Boost credentials. **Note:** Using your own Google Cloud account for this lab may incur extra charges.

5. Click through the subsequent pages:

   - Accept the terms and conditions.
   - Do not add recovery options or two-factor authentication (because this is a temporary account).
   - Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



Activate Cloud Shell

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

1. Click **Activate Cloud Shell** ▶_ at the top of the Google Cloud console.

When you are connected, you are already authenticated, and the project is set to your **PROJECT_ID**. The output contains a line that declares the **PROJECT_ID** for this session:

Your Cloud Platform project in this session is set to YOUR_PROJECT_ID

`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

2. (Optional) You can list the active account name with this command:

```
gcloud auth list
```

3. Click **Authorize**.

4. Your output should now look like this:

Output:

```
ACTIVE: * ACCOUNT: student-01-xxxxxxxxxxxx@qwiklabs.net To set the active account, run: $ gcloud config set account `ACCOUNT`
```

5. (Optional) You can list the project ID with this command:

```
gcloud config list project
```

Output:

```
[core] project = <project_ID>
```

Example output:

[core] project = qwiklabs-gcp-44776a13dea667a6 **Note:** For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Task 1. Set up your environment

1. Set your project:

```
gcloud config set project {{{project_0.project_id | "filled in at lab start"}}}
```

2. Set the Cloud Run region to a supported region:

```
gcloud config set run/region {{{project_0.default_region | "REGION"}}}
```

3. Set the Cloud Run platform default to `managed`:

```
gcloud config set run/platform managed
```

4. Set the location default of Eventarc for Cloud Run:

```
gcloud config set eventarc/location {{{project_0.default_region | "REGION"}}}
```

Task 2. Enable service account

Next, configure a couple of service accounts needed for the Audit Log trigger.

1. Store the Project Number in an environment variable:

```
export PROJECT_NUMBER="$(gcloud projects list \ --filter=$(gcloud config get-value project) \ --format='value(PROJECT_NUMBER)')"
```

2. Grant the `eventarc.admin` role to the default Compute Engine service account:

```
gcloud projects add-iam-policy-binding $(gcloud config get-value project) \ --member=serviceAccount:${PROJECT_NUMBER}-compute@developer.gserviceaccount.com \ --role='roles/eventarc.admin'
```

Click **Check my progress** to verify that you've performed the above task. Enable Service Account.

Task 3. Event discovery

Registered sources and the types of events can be discovered using the command line.

1. To see the list of different types of events, run the following:

```
gcloud beta eventarc attributes types list
```

Output:

```
NAME DESCRIPTION google.cloud.audit.log.v1.written Cloud Audit Log written
google.cloud.pubsub.topic.v1.messagePublished Cloud Pub/Sub message published
```

2. To get more information about each event, run:

```
gcloud beta eventarc attributes types describe \ google.cloud.pubsub.topic.v1.messagePublished
```

Output:

```
attributes: type description: Cloud Pub/Sub message published name:
google.cloud.pubsub.topic.v1.messagePublished
```

Task 4. Create a Cloud Run sink

1. Set up an environment variable for the service:

```
export SERVICE_NAME=event-display
```

2. Set up an environment variable for the image:

```
export IMAGE_NAME="gcr.io/cloudrun/hello"
```

3. Deploy your containerized application to Cloud Run:

```
gcloud run deploy ${SERVICE_NAME} \ --image ${IMAGE_NAME} \ --allow-unauthenticated \ --max-
instances=3
```

On successful deployment, the command line displays the service URL. At this point the service is up and running.

You can now visit your deployed container by opening the service URL in any browser window.

Click **Check my progress** to verify that you've performed the above task. Create a Cloud Run sink.

Task 5. Create a Cloud Pub/Sub event trigger

One way of receiving events is through Cloud Pub/Sub. Custom applications can publish messages to Cloud Pub/Sub and these messages can be delivered to Google Cloud Run sinks via Eventarc for Cloud Run.

Create a trigger

1. First, get more details on the parameters you'll need to construct a trigger for events from Cloud Pub/Sub:

gcloud beta eventarc attributes types describe \ google.cloud.pubsub.topic.v1.messagePublished

2. Create a trigger to filter events published to the Cloud Pub/Sub topic to your deployed Cloud Run service:

gcloud beta eventarc triggers create trigger-pubsub \ --destination-run-service=${SERVICE_NAME} \ --matching-criteria="type=google.cloud.pubsub.topic.v1.messagePublished"

Find the topic

- The Pub/Sub trigger creates a Pub/Sub topic behind the scenes. Find it, and assign it to an environment variable:

  export TOPIC_ID=$(gcloud eventarc triggers describe trigger-pubsub \ --format='value(transport.pubsub.topic)')

Test the trigger

1. You can check that the trigger is created by listing all triggers:

gcloud eventarc triggers list **Note:** You might need to wait for up to 5 minutes for the trigger creation to be propagated and for it to begin filtering events.

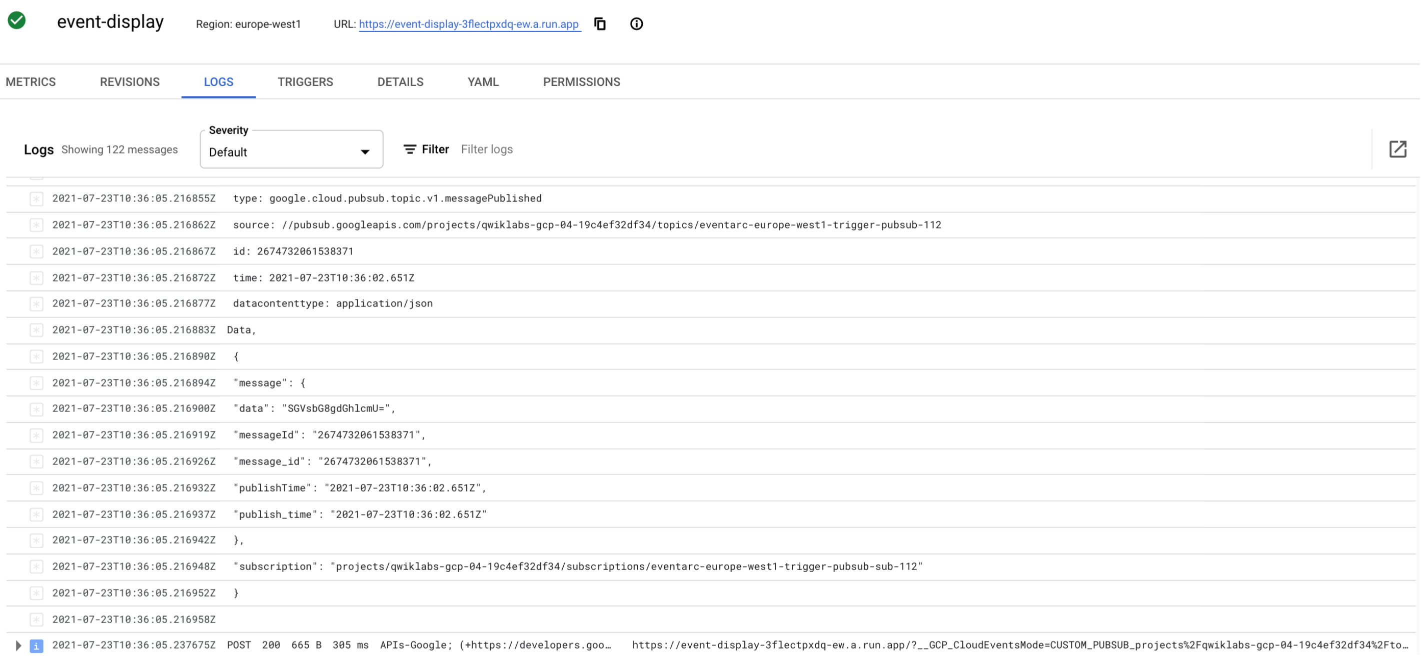
2. In order to simulate a custom application sending message, you can use a `gcloud` command to to fire an event:

gcloud pubsub topics publish ${TOPIC_ID} --message="Hello there"

3. In the **Navigation menu** > **Serverless** > **Cloud Run**, click on **event display**.

4. Click on **Logs**.

The Cloud Run sink you created logs the body of the incoming message. You can view this in the Logs section of your Cloud Run instance:



```
2021-07-23T10:36:05.216855Z   type: google.cloud.pubsub.topic.v1.messagePublished
2021-07-23T10:36:05.216862Z   source: //pubsub.googleapis.com/projects/qwiklabs-gcp-04-19c4ef32df34/topics/eventarc-europe-west1-trigger-pubsub-112
2021-07-23T10:36:05.216867Z   id: 2674732061538371
2021-07-23T10:36:05.216872Z   time: 2021-07-23T10:36:02.651Z
2021-07-23T10:36:05.216877Z   datacontenttype: application/json
2021-07-23T10:36:05.216883Z  Data,
2021-07-23T10:36:05.216890Z   {
2021-07-23T10:36:05.216894Z   "message": {
2021-07-23T10:36:05.216900Z   "data": "SGVsbG8gdGhlcmU=",
2021-07-23T10:36:05.216919Z   "messageId": "2674732061538371",
2021-07-23T10:36:05.216926Z   "message_id": "2674732061538371",
2021-07-23T10:36:05.216932Z   "publishTime": "2021-07-23T10:36:02.651Z",
2021-07-23T10:36:05.216937Z   "publish_time": "2021-07-23T10:36:02.651Z"
2021-07-23T10:36:05.216942Z   },
2021-07-23T10:36:05.216948Z   "subscription": "projects/qwiklabs-gcp-04-19c4ef32df34/subscriptions/eventarc-europe-west1-trigger-pubsub-sub-112"
2021-07-23T10:36:05.216952Z   }
2021-07-23T10:36:05.216958Z
2021-07-23T10:36:05.237675Z  POST  200  665 B  305 ms  APIs-Google; (+https://developers.goo…   https://event-display-3flectpxdq-ew.a.run.app/?__GCP_CloudEventsMode=CUSTOM_PUBSUB_projects%2Fqwiklabs-gcp-04-19c4ef32df34%2Fto…
```

Click **Check my progress** to verify that you've performed the above task. Create a Cloud Pub/Sub event trigger.

Delete the trigger

- You can delete the trigger once you're done testing:

  ```
  gcloud eventarc triggers delete trigger-pubsub
  ```

Task 6. Create a Audit Logs event trigger

Next, set up a trigger to listen for events from Audit Logs. You will listen for Cloud Storage events in Audit Logs.

Create a bucket

1. Create an environment variable for your bucket:

```
export BUCKET_NAME=$(gcloud config get-value project)-cr-bucket
```

2. Create a Cloud Storage bucket in the same region as the deployed Cloud Run service:

```
gsutil mb -p $(gcloud config get-value project) \ -l $(gcloud config get-value run/region) \
gs://${BUCKET_NAME}/
```

Click **Check my progress** to verify that you've performed the above task. Create a bucket.

Enable Audit Logs

In order to receive events from a service, you need to enable audit logs.

1. From the **Navigation menu**, select **IAM & Admin** > **Audit Logs**.

2. In the list of services, check the box for `Google Cloud Storage`.

3. On the right hand side, click the **LOG TYPE** tab. **Admin Write** is selected by default, make sure you also select **Admin Read**, **Data Read**, **Data Write** and then click **Save**.

Test audit logs

To learn how to identify the parameters you'll need to set up an actual trigger, and perform an actual operation.

1. Run the following to create a text file named `random.txt`:

echo "Hello World" > random.txt

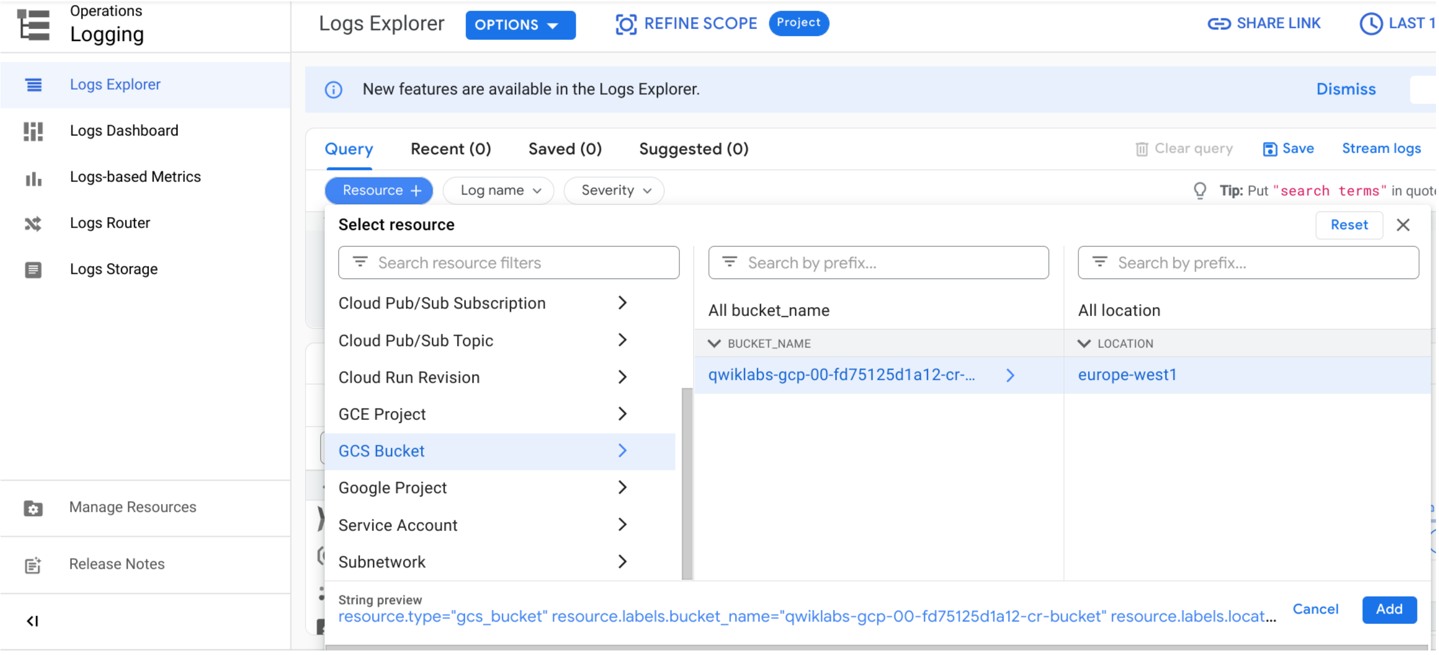
2. Upload the file `random.txt` to the bucket:

gsutil cp random.txt gs://${BUCKET_NAME}/random.txt

Now, see what kind of audit log this update generated.

3. In the Cloud Console, go to **Navigation menu** > **Logging** > **Logs Explorer**.

4. Under **Resource**, choose **GCS Bucket > [Bucket Name] > Location** then choose your bucket and its location. Click **Add**.



New features are available in the Logs Explorer.

Tip: Put "search terms" in quot

Select resource

Cloud Pub/Sub Subscription
Cloud Pub/Sub Topic
Cloud Run Revision
GCE Project
GCS Bucket
Google Project
Service Account
Subnetwork

All bucket_name
BUCKET_NAME
qwiklabs-gcp-00-fd75125d1a12-cr-...

All location
LOCATION
europe-west1

String preview
resource.type="gcs_bucket" resource.labels.bucket_name="qwiklabs-gcp-00-fd75125d1a12-cr-bucket" resource.labels.locat...

Note: There is some latency for audit logs to show up in Logs Viewer UI. If you don't see **GCS Bucket** under the list of resources, wait a little before trying again.

5. Click on **Run Query**.

Once you run the query, you'll see logs for the storage bucket. One of those should be `storage.buckets.create`.

6. Note the `serviceName`, `methodName` and `resourceName`. You will use these in creating the trigger.

Create a trigger

You are now ready to create an event trigger for Audit Logs.

1. Get more details on the parameters you'll need to construct the trigger:

gcloud beta eventarc attributes types describe google.cloud.audit.log.v1.written

2. Create the trigger with the right filters:

gcloud beta eventarc triggers create trigger-auditlog \ --destination-run-service=${SERVICE_NAME} \ --matching-criteria="type=google.cloud.audit.log.v1.written" \ --matching-criteria="serviceName=storage.googleapis.com" \ --matching-criteria="methodName=storage.objects.create" \ --service-account=${PROJECT_NUMBER}-compute@developer.gserviceaccount.com **Note: resourceName parameter**

- There is an optional resourceName field. Providing a complete resource path (eg. projects/_/buckets/test123) will filter for events pertaining to the specific resource. Providing no resource path at all will filter for events for any resource corresponding to the provided serviceName and methodName. Partial resource names (eg. projects/project-id) are not accepted and will malfunction.
- For methodsNames that are of a ‘create' variety (eg. storage.buckets.create for creating a Cloud Storage bucket) resourceNames are best left blank as resourceName may be dynamically generated for some serviceNames and cannot be predicted beforehand.
- For methodNames that are of a ‘read', ‘update' or ‘delete' variety (eg. storage.buckets.update for updating a specific Cloud Storage bucket), you may specify the complete resource path.

Test the trigger

1. List all triggers to confirm that the trigger was successfully created:

gcloud eventarc triggers list

2. Wait for up to 10 minutes for the trigger creation to be propagated and for it to begin filtering events.

Once ready, it will filter create events and send them to the service. You're now ready to fire an event.

3. Upload the same file to the Cloud Storage bucket as you did earlier:

gsutil cp random.txt gs://${BUCKET_NAME}/random.txt

4. Navigate to **Navigation menu** > **Cloud Run** to check the logs of the Cloud Run service, you should see the received event.

Click **Check my progress** to verify that you've performed the above task. Create a Audit Logs event trigger.

Delete the trigger

- You can delete the trigger once done testing:

gcloud eventarc triggers delete trigger-auditlog

Congratulations!

You have successfully learned about Events for Cloud Run on Google Cloud infrastructure. Over the course of this lab, you have performed the following tasks:

- Events for Cloud Run
- Create a Cloud Run sink
- Create an Event trigger for Cloud Pub/Sub
- Create an Event trigger for Audit Logs

Learn more / Next steps

- Eventrac Audit Logging

Google Cloud training and certification

...helps you make the most of Google Cloud technologies. Our classes include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. Certifications help you validate and prove your skill and expertise in Google Cloud technologies.

Manual Last Updated May 11, 2023

Lab Last Tested May 11, 2023