

[Serverless Computing](https://cloud.google.com/products/serverless/) (https://cloud.google.com/products/serverless/)

[Cloud Run: Serverless Computing](https://cloud.google.com/run/) (https://cloud.google.com/run/)

[Documentation](https://cloud.google.com/run/docs/) (https://cloud.google.com/run/docs/) [Guides](#)

Logging and viewing logs

This page describes the logs available when using Cloud Run, and how to view and write logs.

Cloud Run has two types of logs, and these are automatically sent to [Stackdriver Logging](https://cloud.google.com/logging/docs/) (https://cloud.google.com/logging/docs/):

- Request logs: logs of requests sent to Cloud Run services. These logs are created automatically.
- Container logs: logs emitted from the container instances, typically from your own code, written to supported locations as described in [Writing container logs](#) (#container-logs).

Viewing logs

You can view logs for your service in a couple of ways:

- Use the Cloud Run page in the Cloud Console
- Use the Stackdriver Logging page in the Cloud Console.

Both of these viewing methods examine the same logs stored in Stackdriver Logging, but the Stackdriver Logging page provides more details and more filtering capabilities.

Viewing logs in Cloud Run

To view logs in the Cloud Run page:

1. [GO TO CLOUD RUN](https://console.cloud.google.com/run) (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/RUN)
2. Click the desired service in the displayed list.
3. Click the **LOGS** tab to get the request and container logs for all revisions of this service. You can filter by log severity level.

Viewing logs in Stackdriver Logging

To view your Cloud Run logs in the Stackdriver Logging page:

1. Go to the **Stackdriver Logging > Logs** (Logs Viewer) page in the Cloud Console:

[GO TO THE LOGS VIEWER PAGE \(HTTPS://CONSOLE.CLOUD.GOOGLE.COM/LOGS/VIEWER\)](https://console.cloud.google.com/logs/viewer)

2. Select an existing Google Cloud project at the top of the page, or create a new project.
3. Using the drop-down menus, select the resource:
 - **Cloud Run Revision** for Cloud Run (fully managed).
 - **Kubernetes Container** for Cloud Run for Anthos on Google Cloud.

For more information, read [Viewing logs](https://cloud.google.com/logging/docs/view/overview) (https://cloud.google.com/logging/docs/view/overview) on Stackdriver Logging.

Viewing logs using the command line

To view your Cloud Run (fully managed) logs using the command line:

```
gcloud logging read "resource.type=cloud_run_revision AND resource.labels.service_name"
```

Replace **PROJECT-ID** with your Google Cloud project ID. You can view your project ID by running the command `gcloud config get-value project`. Replace **SERVICE** with the name of your Cloud Run service.

Writing container logs

When you write logs from your service, they will be picked up automatically by Stackdriver Logging so long as the logs are written to any of these locations:

- [Standard output \(stdout\) or standard error \(stderr\) streams](https://en.wikipedia.org/wiki/Standard_streams) (https://en.wikipedia.org/wiki/Standard_streams)
- Any files under the `/var/log` directory
- syslog (`/dev/log`)
- Logs written using [Stackdriver Logging Client Libraries](https://cloud.google.com/logging/docs/reference/libraries) (https://cloud.google.com/logging/docs/reference/libraries), which are available for many popular languages

Most developers are expected to write logs using standard output and standard error.

The container logs written to these supported locations are automatically associated with the Cloud Run service, revision, and location. Exceptions contained in these logs are captured by and reported in [Stackdriver Error Reporting](https://cloud.google.com/run/docs/error-reporting) (<https://cloud.google.com/run/docs/error-reporting>).

Using simple text vs structured JSON in logs

When you write logs, you can send a simple text string or send a single line of serialized JSON, also called "structured" data. This is picked up and parsed by Stackdriver Logging and is placed into `jsonPayload`. In contrast, the simple text message is placed in `textPayload`.

Writing structured logs

The following snippet shows how to write structured log entries. It also shows how to correlate log messages with the corresponding request log.

NODE.JSPYTHONGOJAVA

[run/logging-manual/app.js](https://github.com/GoogleCloudPlatform/nodejs-docs-samples/blob/master/run/logging-manual/app.js)
(<https://github.com/GoogleCloudPlatform/nodejs-docs-samples/blob/master/run/logging-manual/app.js>)

GOOGLECLOUDPLATFORM/NODEJS-DOCS-SAMPLES/BLOB/MASTER/RUN/LOGGING-MANUAL/APP.JS

[FEEDBACK \(#\)](#)

```
// Uncomment and populate this variable in your code:
// $project = 'The project ID of your Cloud Run service';

// Build structured log messages as an object.
const globalLogFields = {};

// Add log correlation to nest all log messages beneath request log in Log Viewer.
const traceHeader = req.header('X-Cloud-Trace-Context');
if (traceHeader && project) {
  const [trace] = traceHeader.split('/');
  globalLogFields[
    'logging.googleapis.com/trace'
  ] = `projects/${project}/traces/${trace}`;
```

```
}

// Complete a structured log entry.
const entry = Object.assign(
  {
    severity: 'NOTICE',
    message: 'This is the default display field.',
    // Log viewer accesses 'component' as 'jsonPayload.component'.
    component: 'arbitrary-property',
  },
  globalLogFields
);

// Serialize to a JSON string and output.
console.log(JSON.stringify(entry));
```

You can use the [Stackdriver Logging Client Libraries](https://cloud.google.com/logging/docs/reference/libraries)

(<https://cloud.google.com/logging/docs/reference/libraries>) to get this functionality automatically.

Special JSON fields in messages

When you provide a structured log as a JSON dictionary, some special fields are stripped from the `jsonPayload` and are written to the corresponding field in the generated `LogEntry`.

(<https://cloud.google.com/logging/docs/reference/v2/rest/v2/LogEntry>) as described in the documentation for [special fields](#)

(<https://cloud.google.com/logging/docs/agent/configuration#special-fields>).

For example, if your JSON includes a `severity` property, it is removed from the `jsonPayload` and appears instead as the log entry's `severity`. The `message` property is used as the main display text of the log entry if present. For more on special properties read the [Logging Resource](#) ([#log-resource](#)) section below.

Correlating your container logs with a request log

In the Stackdriver logs viewer, logs correlated by the same `trace` are viewable in "parent-child" format: when you click on the triangle icon at the left of the request log entry, the container logs related to that request show up nested under the request log.

Container logs are not automatically correlated to request logs unless you use a [Stackdriver Logging client library](#) (<https://cloud.google.com/logging/docs/reference/libraries>). To correlate container logs with request logs without using a client library, you can use a structured JSON log line that contains a `logging.googleapis.com/trace` field with the trace identifier extracted from the `X-Cloud-Trace-Context` header as shown in the above sample for [structured logging](#) (`#writing_structured_logs`).

Controlling request log resource usage

Request logs are created automatically. Although you cannot control the amount of request logs directly from Cloud Run, you can make use of the [logs exclusion](#) (<https://cloud.google.com/logging/docs/exclusions#overview>) feature from Stackdriver Logging.

Logging resource

Clicking on a log entry in the Stackdriver Logs Viewer opens up a JSON formatted log entry so you can drill down to the details you want.

All of the fields in a log entry, such as timestamps, severity, and `HttpRequest` are standard, and are described in the documentation for a [log entry](#).

(<https://cloud.google.com/logging/docs/reference/v2/rest/v2/LogEntry>).

However, there are some labels or resource labels that are special to Cloud Run. These are listed here with sample contents:

```
{
  httpRequest: {...}
  insertId: "5c82b3d1000ece0000000000"
  labels: {
    instanceId: "00bf4bf0000fb59c906a0000c9e29c2c4e06dce9150000000056008d2b6460f16"
  }
  logName: "projects/my-project/logs/run.googleapis.com%2Frequests"
  receiveTimestamp: "2019-03-08T18:26:25.981686167Z"
  resource: {
    labels: {
      configuration_name: "myservice"
      location: "us-central1"
      project_id: "my-project"
    }
  }
}
```

```
  revision_name: "myservice-00002"
  service_name: "myservice"
}
type: "cloud_run_revision"
}
severity: "INFO"
timestamp: "2019-03-08T18:26:25.970397Z"
}
```

Field	Values and notes
instanceId	The container instance that handled the request.
logName	Identifies the log, for example, request log, standard error, standard output, etc.
configuration_name	The Configuration resource that created the revision that served the request.
location	Identifies the GCP location of the service.
project_id	The project the service is deployed to.
revision_name	The revision that served the request.
service_name	The service that served the request.
type	cloud_run_revision . The Cloud Run resource type.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated December 12, 2019.