You can write logs to Logging from Python applications by using the Python logging handler included with the Logging client library, or by using Stackdriver Logging API Cloud client library for Python directly.

1. Sign in (https://accounts.google.com/Login) to your Google Account.

   If you don't already have one, sign up for a new account (https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a Cloud project.

★ **Note**: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

   Go to the project selector page (https://console.cloud.google.com/projectselector2/home/dashboard)

3. Make sure that billing is enabled for your Google Cloud project. Learn how to confirm billing is enabled for your project (/billing/docs/how-to/modify-project).

4. Enable the Stackdriver Logging API.
   Enable the API (https://console.cloud.google.com/flows/enableapi?apiid=logging.googleapis.com)

5. Prepare your environment for Python development.
   Go to the Python setup guide (/python/setup)

To install the Stackdriver Logging library for Python, see Installing the client library (/logging/docs/reference/libraries#client-libraries-install-python).

Once installed, this library includes logging handlers to connect Python's standard logging module to Stackdriver, as well as an API client library to access Stackdriver Logging manually.

To send all log entries to Stackdriver by attaching the Stackdriver Logging handler to the Python root logger, use the `setup_logging` helper method:

logging/cloud-client/handler.py
(https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/logging/cloud-client/handler.py)

```
ıb (https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/logging/cloud-client/handler.py)
```

Once the handler is attached, any logs at, by default, `INFO` level or higher which are emitted in your application will be sent to Logging:

logging/cloud-client/handler.py
(https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/logging/cloud-client/handler.py)

```
ıb (https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/logging/cloud-client/handler.py)
```

If messages are logged to Logging from App Engine or Google Kubernetes Engine, the handler will send them to those environments' respective resource types; otherwise, logs will by default appear under the `python` log in the `Global` resource type.

To attach the Stackdriver logging handler to only select Python loggers, or to otherwise configure the logging handler, go to the API library documentation (https://googleapis.dev/python/logging/latest/stdlib-usage.html).

For more information on installation, see the documentation (https://googleapis.github.io/google-cloud-python/latest/logging/usage.html) for the Stackdriver Logging library for Python. You can also report issues using the issue tracker (https://github.com/GoogleCloudPlatform/google-cloud-python/issues).

For information on using the Stackdriver Logging Cloud client library for Python directly, see Stackdriver Logging Client Libraries (/logging/docs/reference/libraries).

Using Stackdriver Logging library for Python requires the Cloud IAM (/iam/docs/understanding-roles) Logs Writer role (/logging/docs/access-control#permissions_and_roles) on Google Cloud. Most Google Cloud environments provide this role by default.

App Engine (/appengine/docs/) grants the Logs Writer role (/logging/docs/access-control#permissions_and_roles) by default.

The Stackdriver Logging library for Python can be used without needing to explicitly provide credentials.

Stackdriver Logging is automatically enabled for App Engine applications. No additional setup is required.

Logs written to **`stdout`** and **`stderr`** are automatically sent to Stackdriver Logging for you, without needing to use Stackdriver Logging library for Python.

On Google Kubernetes Engine (/kubernetes-engine/docs/), you must add the **`logging.write`** access scope when creating the cluster:

To use the Stackdriver Logging library for Python on a **Compute Engine** VM instance, you do not need to install the Stackdriver Logging agent.

When using Compute Engine (/compute/docs/) VM instances, add the **`cloud-platform`** access scope to each instance. When creating a new instance through the Google Cloud Console, you can do this in the **Identity and API access** section of the **Create Instance** panel. Use the Compute Engine default service account or another service account of your choice, and select **Allow full access to all Cloud APIs** in the **Identity and API access** section. Whichever service account you select, ensure that it has been granted the Logs Writer role (/logging/docs/access-control#permissions_and_roles) in the **IAM & admin** section of the Cloud Console.

To use the Stackdriver Logging library for Python outside of Google Cloud, including running the library on your own workstation, on your data center's computers, or on the VM instances of another cloud provider, you must supply your Google Cloud project ID and appropriate service account credentials directly to the Stackdriver Logging library for Python.

You can create and obtain service account credentials manually (/docs/authentication/production#obtaining_and_providing_service_account_credentials_manually). When specifying the **Role** field, use the Logs Writer role (/logging/docs/access-control#permissions_and_roles). For more information on Cloud Identity and Access Management roles, go to Access control guide (/logging/docs/access-control).

After deployment, you can view the logs in the Cloud Console Logs Viewer.

Go to the Logs Viewer (https://console.cloud.google.com/logs/viewer)

In the Logs Viewer, you must specify one or more resources, but the resource selection might not be obvious. Here are some tips to help you get started:

- If you are deploying your application to App Engine or using the App Engine-specific libraries, set your resource to **GAE Application**.
- If you are deploying your application on Compute Engine, set the resource to **GCE VM Instance**.
- If you are deploying your application on Google Kubernetes Engine, your cluster's logging configuration determines the resource type of the log entries. For a detailed discussion on the Legacy Stackdriver and the Stackdriver Kubernetes Monitoring solutions, and how those options affect the resource type, see Migrating to Stackdriver Kubernetes Monitoring (/monitoring/kubernetes-engine/migration#stackdriver-options).
- If your application is using the Stackdriver Logging API directly, the resource is dependent on the API and your configuration. For example, in your application, you can specify a resource or use a default resource.
- If you don't see any logs in the Logs Viewer, to see all log entries, switch to the advanced query mode and use an empty query.

    1. To switch to the advanced query mode, click **menu** (▾) at the top of the Logs Viewer and then select **Convert to advanced filter**.
    2. Clear the content that appears in the filter box.
    3. Click **Submit Filter**.

    You can examine the individual entries to identify your resources.

For additional information, see Viewing Logs (/logging/docs/view/overview) and Advanced logs queries (/logging/docs/view/advanced-queries).