Setting up on Compute Engine

You can send errors from your Compute Engine applications to Error Reporting in one of two ways:

- By logging to Cloud Logging (#using_logging). If you're already using Cloud Logging, the only additional requirement is that your log entries be recognizable by Error Reporting. For more information on error formatting requirements, read Formatting errors in Cloud Logging (/error-reporting/docs/formatting-error-messages).
- Using the Error Reporting API (#using_the_api). Your application can send HTTP requests using the REST API, or can make use of experimental libraries in several languages.

Using Logging to report errors

To connect your Compute Engine applications to Error Reporting, send your exceptions or other errors to Logging, using a dedicated log name (/logging/docs/basic-concepts#logs) to keep your errors separate from your other logs.

For example:

1. Enable the Cloud Logging API.

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

2. Install the Logging agent that is appropriate for your environment. For instructions, go to Installing the Logging agent (/logging/docs/agent/installation) `google-fluentd`.
3. Modify your application so that it logs exceptions and their stack traces to Logging. Choose a log name for your error information to keep it separate from other logged information.

   You must include all the information for a single error or exception in the same log entry, including all the frames of any stack trace. If all the information is not together, Error Reporting might not pick up the information. You can use the structured JSON format (/error-reporting/docs/formatting-error-messages#json_representation) for your log entry payloads to include different kinds of information for each error.

Add the following to your `pom.xml` file:

compute/error-reporting/pom.xml (https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/compute/error-reporting/pom.xml)

ɔs://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/compute/error-reporting/pom.xml)

```
<dependency>
  <groupId>org.fluentd</groupId>
  <artifactId>fluent-logger</artifactId>
  <version>0.3.4</version>
</dependency>
```

Then use code like the following to send the exception data:

compute/error-reporting/src/main/java/com/example/compute/errorreporting/ExceptionUtil.java (https://github.com/GoogleCloudPlatform/java-docs-samples/blob/master/compute/error-reporting/src/main/java/com/example/compute/errorreporting/ExceptionUtil.java)

'master/compute/error-reporting/src/main/java/com/example/compute/errorreporting/ExceptionUtil.java)

```
public class ExceptionUtil {
  private static FluentLogger ERRORS = FluentLogger.getLogger("myapp");

  public static void main(String[] args) {
    try {
      throw new Exception("Generic exception for testing Stackdriver");
    } catch (Exception e) {
      report(e);
    }
  }

  public static void report(Throwable ex) {
    StringWriter exceptionWriter = new StringWriter();
    ex.printStackTrace(new PrintWriter(exceptionWriter));
    Map<String, Object> data = new HashMap<>();
    data.put("message", exceptionWriter.toString());
    Map<String,String> serviceContextData = new HashMap<>();
    serviceContextData.put("service", "myapp");
    data.put("serviceContext", serviceContextData);
    // ... add more metadata
```

```
    ERRORS.log("errors", data);
  }
}
```

Using the Error Reporting API to write errors

The Error Reporting API provides a `report` endpoint for writing error information to the service.

1. Enable the Error Reporting API.

   Enable the API (https://console.cloud.google.com/flows/enableapi?apiid=clouderrorreporting.googlea

2. Report errors to the API using either the REST API or a client library.

   - REST API

     See the reference documentation (/error-reporting/reference/rest/v1beta1/projects.events/report) for information about the API.

   - Using client libraries

     Libraries exist in a limited number of languages to help you call the Error Reporting API from your application:

     - ASP.NET (http://googleapis.github.io/google-cloud-dotnet/docs/Google.Cloud.Diagnostics.AspNet/index.html)
     - Go (/error-reporting/docs/setup/go)
     - Java (/error-reporting/docs/setup/java)
     - Node.js (/error-reporting/docs/setup/nodejs)
     - Ruby (/error-reporting/docs/setup/ruby)
     - Python (/error-reporting/docs/setup/python)

Samples

ASP.NET C# (#c) Go (#go) Java (#java) Node.js (#node.js) Ruby (#ruby) Python (#python)

The Stackdriver ASP.NET NuGet package (http://googleapis.github.io/google-cloud-dotnet/docs/Google.Cloud.Diagnostics.AspNet/index.html) reports uncaught exceptions from ASP.NET web applications to Error Reporting.

Install the NuGet package

To install the Stackdriver ASP.NET NuGet package in Visual Studio:

1. Right-click your solution and select **Manage NuGet packages for solution**.
2. Select the **Include prerelease** checkbox.
3. Search for and install the package named `Google.Cloud.Diagnostics.AspNet`.

Usage

Once you've installed the Stackdriver ASP.NET NuGet package, add the following statement to your application code to start sending errors to Stackdriver:

```
using Google.Cloud.Diagnostics.AspNet;
```

Add the following `HttpConfiguration` code to the `Register` method of your .NET web app (replacing `your-project-id` with your actual project ID (https://support.google.com/cloud/answer/6158840?hl=en) to enable the reporting of exceptions:

error-reporting/api/diagnostics/App_Start/WebApiConfig.cs (https://github.com/GoogleCloudPlatform/dotnet-docs-samples/blob/vs2015/error-reporting/api/diagnostics/App_Start/WebApiConfig.cs)

IPlatform/dotnet-docs-samples/blob/vs2015/error-reporting/api/diagnostics/App_Start/WebApiConfig.cs)

```
public static void Register(HttpConfiguration config)
{
    string projectId = "YOUR-PROJECT-ID";
    string serviceName = "NAME-OF-YOUR-SERVICE";
    string version = "VERSION-OF-YOUR-SERVCICE";
    // ...
    // Add a catch all for the uncaught exceptions.
    config.Services.Add(typeof(IExceptionLogger),
```

```
        ErrorReportingExceptionLogger.Create(projectId, serviceName, version));
    // ...
}
```

Once you've added this method to your ASP.NET application, you can view any uncaught exceptions that occur as they get reported to Google Cloud in the Error Reporting (https://console.cloud.google.com/errors) section of the Cloud Console.

Last updated 2020-08-14 UTC.