

fn-google-cloud-dlp Functions for IBM Resilient

- [fn-google-cloud-dlp Functions for IBM Resilient](#)
 - [Overview](#)
 - [Key Features](#)
 - [Requirements](#)
 - [Authenticating to Google Cloud](#)
 - It is recommended to create a new service or user account with the [DLP.User](#) permission. You will then be given a keyfile which you can set as the `GOOGLE_APPLICATION_CREDENTIALS` bash value which will be the absolute path to your Keyfile
 - [Installation](#)
 - [Uninstall](#)
 - [Troubleshooting](#)
 - [Resilient Action Status](#)
 - [Resilient Scripting Log](#)
 - [Resilient Logs](#)
 - [Resilient-Circuits](#)
 - [Support](#)

Overview

Resilient Circuits Components for 'fn_google_cloud_dlp'

The screenshot shows the IBM Resilient console interface. At the top is a navigation bar with the 'Resilient' logo and tabs for Dashboards, Simulations, Incidents, and a 'Create' button. A user profile for 'Alfred Penny... WayneCorp' is visible on the right. Below the navigation bar is a 'Customization Settings' section with a horizontal menu containing Layouts, Rules, Scripts, Workflows (selected), Functions, Message Destinations, Phases & Tasks, Incident Types, Breach, and Artifacts. The 'Workflows' section is active, displaying a table of workflows. The table has columns for Workflow Name, Description, Object Type, and Rules. Three example workflows are listed, all involving Google Cloud DLP integration for de-identifying or inspecting PII. A 'New Workflow' button is located in the top right of the Workflows section.

Workflow Name	Description	Object Type	Rules
Example: Google Cloud DLP - De-Identify Artifact	An example workflow ran at an attachment level that sends the artifact data to Google Cloud's DLP Service and aims to de-identify the types of personal information specified. By default 14 types are selected out of 50+ types. The return result is a new artifact with the PII removed.	Artifact	Example: Google Cloud - Remove PII from String
Example: Google Cloud DLP - De-Identify Attachment	An example workflow ran at an attachment level that sends the attachment data to Google Cloud's DLP Service and aims to de-identify the types of personal information specified. By default 14 types are selected out of 50+ types. The return result is a new attachment with the PII removed.	Attachment	Example: Google Cloud - Remove PII from Attachment
Example: Google Cloud DLP - Inspect Attachment for PII	An example workflow ran at an attachment level that sends the attachment data to Google Cloud's DLP Service and aims to de-identify the types of personal information specified. By default 14 types are selected out of 50+ types. Returned results include a list of findings generated from the input including the finding itself, what type of info it was matched against and the likelihood that the match is accurate.	Attachment	Example: Google Cloud - Inspect Attachment for PII

The Resilient Integration with Google Cloud DLP provides tools to integrate into your Incident Response Plan. The integration brings Automation and Orchestration capabilities for either identifying, redacting or de-identifying Personally identifiable information (PII) in a body of text.

Key Features

- Inspect a text-based attachment for Personal Identifiable Information
 - Search for and redact Personal Identifiable Information from an attachment or artifact
-

Requirements

- IBM Resilient >= **v31.0.4235**
- An Integrations Server running **resilient-circuits** >= **v31.0.0**
 - To setup an Integrations Server see: ibm.biz/res-int-server-guide

Authenticating to Google Cloud

Application Default Credentials: Application Default Credentials are the preferred way to authenticate when using a client library to interface with Google Cloud.

Services using ADC look for credentials within a `GOOGLE_APPLICATION_CREDENTIALS` environment variable. Unless you specifically want to have ADC use other credentials (for example, user credentials), set this environment variable to point to your service account key file. It is recommended to create a new service or user account with the **DLP.User** permission. You will then be given a keyfile which you can set as the `GOOGLE_APPLICATION_CREDENTIALS` bash value which will be the absolute path to your Keyfile

Installation

- Download the **fn_google_cloud_dlp.zip**
- Copy the **.zip** to your Integrations Server and SSH into it.
- **Unzip** the package:

```
$ unzip fn_google_cloud_dlp-x.x.x.zip
```

- **Install** the package:

```
$ pip install fn_google_cloud_dlp-x.x.x.tar.gz
```

- Import the **configurations** into your app.config file:

```
$ resilient-circuits config -u
```

- Import the `fn_google_cloud_dlp` **customizations** into the Resilient Appliance:

```
$ resilient-circuits customize -y -l fn-google-cloud-dlp
```

- Open the config file, scroll to the bottom and edit your `fn_google_cloud_dlp` **configurations**:

```
$ nano ~/.resilient/app.config
```

Config	Required	Example	Description
gcp_project	Yes	<YOUR_GOOGLE_PROJECT_ID>	The Google Cloud Project that will be used with DLP
gcp_dlp_masking_char	Yes	#	What character will be used to mask PII

- **Save** and **Close** the `app.config` file.
- [Optional]: Run **selftest** to test you the Integration is configured:

```
$ resilient-circuits selftest -l fn-google-cloud-dlp
```

- **Run** `resilient-circuits` or restart the Service on Windows/Linux:

```
$ resilient-circuits run
```

Uninstall

- SSH into your Integrations Server
- **Uninstall** the package:

```
$ pip uninstall fn-google-cloud-dlp
```

- Open the config file, scroll to the `[fn_google_cloud_dlp]` section and remove the section or prefix `#` to comment out the section.
- **Save** and **Close** the `app.config` file.

Troubleshooting

There are several ways to verify the successful operation of a function.

Resilient Action Status

- When viewing an incident, use the Actions menu to view **Action Status**.
- By default, pending and errors are displayed.
- Modify the filter for actions to also show Completed actions.
- Clicking on an action displays additional information on the progress made or what error occurred.

Resilient Scripting Log

- A separate log file is available to review scripting errors.
- This is useful when issues occur in the pre-processing or post-processing scripts.
- The default location for this log file is: `/var/log/resilient-scripting/resilient-scripting.log`.

Resilient Logs

- By default, Resilient logs are retained at `/usr/share/co3/logs`.
- The `client.log` may contain additional information regarding the execution of functions.

Resilient-Circuits

- The log is controlled in the `.resilient/app.config` file under the section [resilient] and the property `logdir`.
- The default file name is `app.log`.
- Each function will create progress information.
- Failures will show up as errors and may contain python trace statements.

Support

Name	Version	Author	Support URL
fn_google_cloud_dlp	1.0.0	Ryan	http://ibm.biz/resilientcommunity