ScienceLogic

Google Cloud Platform Integrations

Google Cloud Platform SyncPack Version 1.0.0

Google Cloud Automation PowerPack Version 100

Table of Contents

Chapter

1

Introduction to the Google Cloud Platform SyncPack

Overview

This chapter describes how you can use the "Google Cloud Platform" SyncPack to automatically synchronize Google Cloud tasks, errors, and messages and SL1 events between your Google Cloud Platform and SL1 systems. The integration is a unidirectional ingestion of SL1 events into the Google Cloud Platform for troubleshooting and remediation.

This chapter covers the following topics:

What Can I Do with this SyncPack?

The "Google Cloud Platform" SyncPack lets you sync Google Cloud tasks, errors, and messages and SL1 events between your Google Cloud Platform and SL1 systems. The integration is unidirectional from SL1 to Google Cloud Platform.

This SyncPack includes the following applications:

- ***Close SL1 Event Delete GCP Task***. This application resolves Google Cloud Platform tasks that were resolved by corresponding SL1 events.

- ***Resolve GCP Event From SL1***. This application resolves SL1 events that were resolved by corresponding Google Cloud Platform events.
- ***Send SL1 Event To GCP Build*** . This application collects event information from SL1 and creates an event in Google Cloud Platform.
- ***Send SL1 Event To GCP Error*** . This application collects event information from SL1 and creates an error in Google Cloud Platform.
- ***Send SL1 Event To GCP Task*** . This application collects event information from SL1 and creates a task in Google Cloud Platform.
- ***Send SL1 Event To GCP Pubsub***. This application collects event information from SL1 and creates a message in Google Cloud Platform Pub/Sub.

For more information about how to configure these applications, see *Configuring Applications for the Google Cloud Platform Synchronization PowerPack*.

Contents of the SyncPack

This section lists the contents of the *Google Cloud Platform* SyncPack.

PowerFlow Applications

- ***Close SL1 Event Delete GCP Task***. This application resolves Google Cloud Platform tasks that were resolved by corresponding SL1 events.
- ***Resolve GCP Event From SL1***. This application resolves SL1 events that were resolved by corresponding Google Cloud Platform events.
- ***Send SL1 Event To GCP Build*** . This application collects event information from SL1 and creates an event in Google Cloud Platform.
- ***Send SL1 Event To GCP Error*** . This application collects event information from SL1 and creates an error in Google Cloud Platform.
- ***Send SL1 Event To GCP Task*** . This application collects event information from SL1 and creates a task in Google Cloud Platform.
- ***Send SL1 Event To GCP Pubsub***. This application collects event information from SL1 and creates a message in Google Cloud Platform Pub/Sub.

For more information about how to configure these applications, see *Configuring Applications for the Google Cloud Platform Synchronization PowerPack*.

Configuration Object

- ***GCP Base Config***. This configuration object can be used as a template after the SyncPack is installed on the PowerFlow system. The configuration object includes the following:

    - Details for connecting to SL1, including the host, username, and password.
    - Details for connecting to Google Cloud Platform, including the topic and subscription IDs, build steps, and task location, queue, and URL.
    - Details for connecting and customizing events and tasks, errors, and messages between SL1 and Google Cloud Platform, including timeout, deadline, number of messages, and message template details.

Steps

The following steps are included in this SyncPack:

- Acknowledge Event In GCP
- Create Task In GCP
- Delete Task In GCP
- Get Event Details From SL1 Publish To GCP
- Post Update to SL1
- Send Event Details As Build To GCP
- Send Event Details As Error To GCP

Prerequisites for this SyncPack

This SyncPack requires the following:

- "Base Steps" SyncPack version 1.3.2 or later
- Google Cloud Platform administrator access to the Administration Portal
- Google Cloud Platform administrator access to the GUI Portal

The following table lists the port access required by PowerFlow and this SyncPack:

Source IP	PowerFlow Destination	PowerFlow Source Port	Destination Port	Requirement
PowerFlow	SL1 API	Any	TCP 443	SL1 API Access
PowerFlow	Google Cloud Platform API	Any	TCP 443	Google Cloud Platform API Access

Installing the SyncPack

A SyncPack file has the **.whl** file extension type. You can download the SyncPack file from the ScienceLogic Support site.

Downloading the SyncPack

To locate and download the SyncPack:

1. Go to the ScienceLogic Support Site.
2. Click the **[Product Downloads]** tab and select *PowerPacks*.
3. In the ***Search PowerPacks*** field, search for this SyncPack and select it from the search results. The **Release Version** page appears.
4. On the **[PowerPack Versions]** tab, click the name of the SyncPack version that you want to install. The **Release File Details** page appears.
5. Click the **[Download File]** button or click the name of the **.zip** file containing the **.whl** file for this SyncPack to start downloading the file.

NOTE: After you download a SyncPack, you can import it to your PowerFlow system using the PowerFlow user interface.

NOTE: If you are installing or upgrading to the latest version of this SyncPack in an offline deployment, see "Installing or Upgrading in an Offline Environment" in the SyncPack release notes to ensure you install any external dependencies.

Importing the SyncPack

To import a SyncPack in the PowerFlow user interface:

1. On the **SyncPacks** page (◎) of the PowerFlow user interface, click **[Import SyncPack]**. The **Import SyncPack** page appears.
2. Click **[Browse]** and select the **.whl** file for the SyncPack you want to install. You can also drag and drop a **.whl** file to the **Import SyncPack** page.
3. Click **[Import]**. PowerFlow registers and uploads the SyncPack. The SyncPack is added to the **SyncPacks** page.
4. You will need to activate and install the SyncPack in PowerFlow. For more information, see ***Activating and Installing a SyncPack***.

NOTE: You cannot edit the content package in a SyncPack published by ScienceLogic. You must make a copy of a ScienceLogic SyncPack and save your changes to the new SyncPack to prevent overwriting any information in the original SyncPack when upgrading.

Installing the SyncPack

To activate and install a SyncPack in the PowerFlow user interface:

1. On the **SyncPacks** page of the PowerFlow user interface, click the **[Actions]** button (⋮) for the SyncPack you want to install and select *Activate & Install*. The **Activate & Install SyncPack** modal appears.

   > **NOTE:** If you try to activate and install a SyncPack that is already activated and installed, you can choose to "force" installation across all the nodes in the PowerFlow system.

   > **TIP:** If you do not see the PowerPack that you want to install, click the Filter icon (☰) on the **SyncPacks** page and select *Toggle Inactive SyncPacks* to see a list of the imported PowerPacks.

2. Click **[Yes]** to confirm the activation and installation. When the SyncPack is activated, the **SyncPacks** page displays a green check mark icon (✅) for that SyncPack. If the activation or installation failed, then a red exclamation mark icon (❗) appears.
3. For more information about the activation and installation process, click the check mark icon (✅) or the exclamation mark icon (❗) in the **Activated** column for that SyncPack. For a successful installation, the "Activate & Install SyncPack" application appears, and you can view the Step Log for the steps. For a failed installation, the **Error Logs** window appears.
4. If you have other versions of the same SyncPack on your PowerFlow system, you can click the **[Actions]** button (⋮) for that SyncPack and select *Change active version* to activate a different version other than the version that is currently running.

Chapter

2

Configuring Applications for the Google Cloud Platform SyncPack

Overview

This chapter describes how to set up the PowerFlow applications for the *Google Cloud Platform* SyncPack.

This chapter covers the following topics:

Configuring the Google Cloud Platform

To configure the Google Cloud Platform to work with PowerFlow, you must do the following:

1. ***Create a Project in Google Cloud***. Give the project a name and location, and take note of the Project ID.
2. ***Create a Service Account and Credentials***. Give the service account a name and ID, and select *Owner* for the Role. In the newly-created service account, go to the **Keys** section and create a key with type *JSON*.
3. ***Configure a Google Cloud Pubsub Topic and Subscription***. In the **Pubsub** page, create a new topic with *Add a default description* selected.
4. ***Enable the Cloud Tasks API***. In the **Cloud Tasks** page, click the **[Enable]** button.
5. ***Create a Cloud Tasks Push Queue***. In the **Cloud Tasks** page, click ***Create Push Queue***. Give the queue a name, select a region from the drop-down menu, and then click **[Create]**.
6. ***Enable the Error Reporting API***. In the **Error Reporting** page, click ***Set Up Error Reporting*** to be directed to the **Guides** page. In the **Guides** page, in the ***Setup Guide*** section click *Error Reporting REST API*. In the Reference page, click **[Enable the API]**.

Creating and Aligning a Configuration Object in PowerFlow

A ***configuration object*** supplies the login credentials and other required information needed to execute the steps for a PowerFlow application. The **Configurations** page (⚙) of the PowerFlow user interface lists all available configuration objects for that system.

You can create as many configuration objects as you need. A PowerFlow application can only use one configuration object at a time, but you can use (or "align") the same configuration object with multiple applications.

To use this SyncPack, you will need to use an existing configuration object in the PowerFlow user interface or create a new configuration object. Next, you need to align that configuration object to the relevant applications.

Creating a Configuration Object

For this SyncPack, you can make a copy of the "GCP Base Config" configuration object, which is the sample configuration file that was installed with the *Google Cloud Platform* SyncPack.

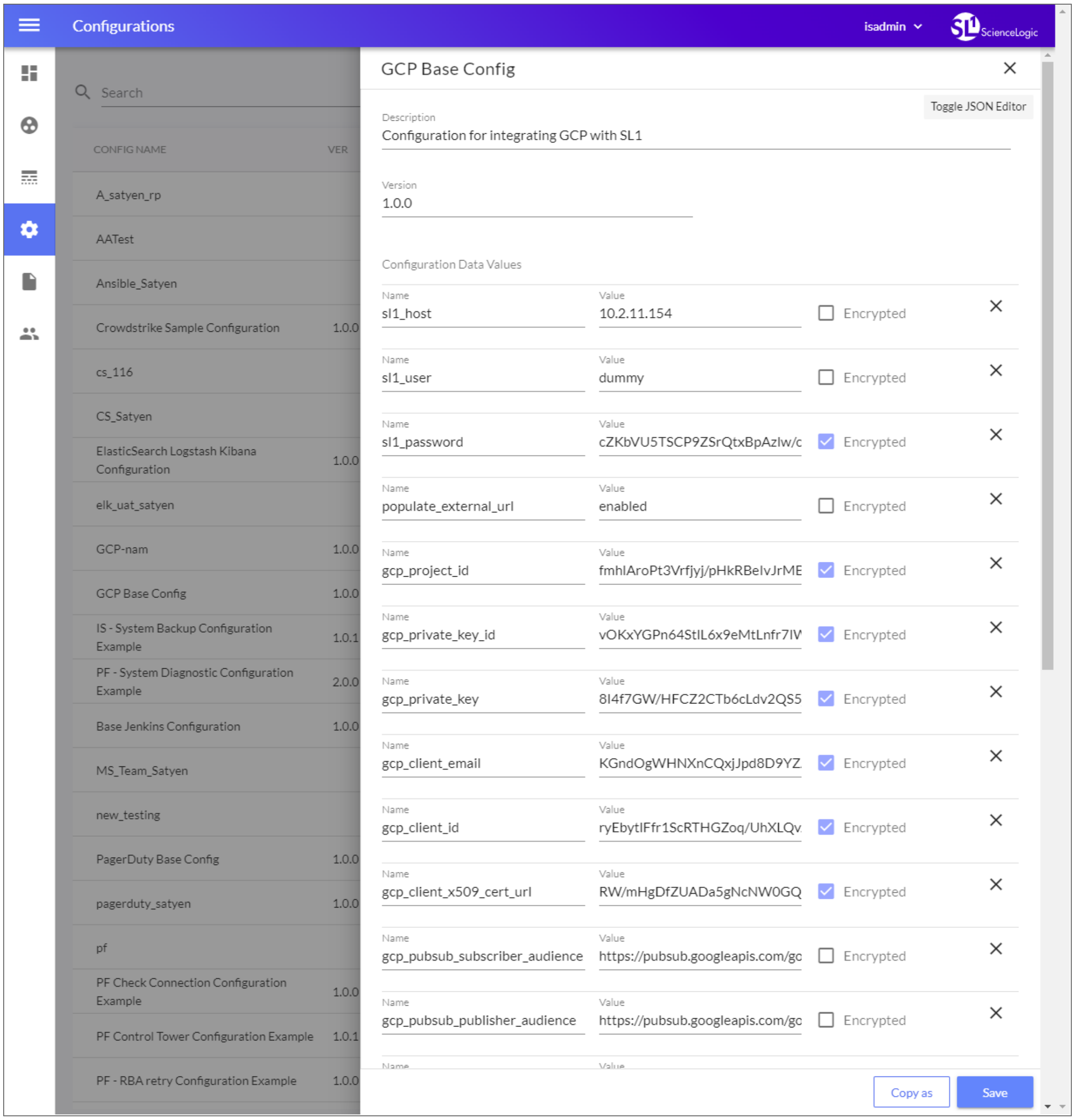
TIP: The "GCP Base Config" configuration object contains all of the required variables. Simply update the variables from that object to match your SL1 and Google Cloud Platform settings.

NOTE: For more information about the Google Cloud Platform terms and concepts in this section, see the Google Cloud Platform documentation.

To create a configuration object based on the "GCP Base Config" configuration object:

1. In the PowerFlow user interface, go to the **Configurations** page (⚙).

2. Click the **[Actions]** button (⋮) for the "GCP Base Config" configuration object and select *Edit*. The **Configuration** pane appears:



CONFIG NAME	VER
A_satyen_rp	
AATest	
Ansible_Satyen	
Crowdstrike Sample Configuration	1.0.0
cs_116	
CS_Satyen	
ElasticSearch Logstash Kibana Configuration	1.0.0
elk_uat_satyen	
GCP-nam	1.0.0
GCP Base Config	1.0.0
IS - System Backup Configuration Example	1.0.1
PF - System Diagnostic Configuration Example	2.0.0
Base Jenkins Configuration	1.0.0
MS_Team_Satyen	
new_testing	
PagerDuty Base Config	1.0.0
pagerduty_satyen	1.0.0
pf	
PF Check Connection Configuration Example	1.0.0
PF Control Tower Configuration Example	1.0.1
PF - RBA retry Configuration Example	1.0.0

GCP Base Config ✕

Toggle JSON Editor

Description
Configuration for integrating GCP with SL1

Version
1.0.0

Configuration Data Values

Name	Value		
sl1_host	10.2.11.154	☐ Encrypted	✕
sl1_user	dummy	☐ Encrypted	✕
sl1_password	cZKbVU5TSCP9ZSrQtxBpAzlw/c	☑ Encrypted	✕
populate_external_url	enabled	☐ Encrypted	✕
gcp_project_id	fmhlAroPt3Vrfjyj/pHkRBelvJrME	☑ Encrypted	✕
gcp_private_key_id	vOKxYGPn64StlL6x9eMtLnfr7IV	☑ Encrypted	✕
gcp_private_key	8I4f7GW/HFCZ2CTb6cLdv2QS5	☑ Encrypted	✕
gcp_client_email	KGndOgWHNXnCQxjJpd8D9YZ	☑ Encrypted	✕
gcp_client_id	ryEbytlFfr1ScRTHGZoq/UhXLQv	☑ Encrypted	✕
gcp_client_x509_cert_url	RW/mHgDfZUADa5gNcNW0GQ	☑ Encrypted	✕
gcp_pubsub_subscriber_audience	https://pubsub.googleapis.com/go	☐ Encrypted	✕
gcp_pubsub_publisher_audience	https://pubsub.googleapis.com/go	☐ Encrypted	✕

3. Click **[Copy as]**. The **Create Configuration** pane appears.
4. Complete the following fields:

   - ***Friendly Name***. Type a name for the configuration object that will display on the **Configurations** page.
   - ***Description***. Type a brief description of the configuration object.
   - ***Author***. Type the user or organization that created the configuration object.
   - ***Version***. Type a version of the configuration object.

5. In the ***Configuration Data*** field, update the default variable definitions to match your PowerFlow configuration:

   - ***sl1_host***. Type the hostname or IP address of your SL1 system.
   - ***sl1_user***. Type the username for your SL1 system.
   - ***sl1_password***. Type the password for your SL1 system.
   - ***populate_external_url***. Type 'enabled' or 'disabled' to add a Google Cloud Platform URL to the corresponding SL1 event. The default value is 'enabled'.
   - ***gcp_project_id***. The unique ID associated with a Google Cloud Platform project. This value is filled in using the "project_id" parameter in the JSON credentials file.
   - ***gcp_private_key_id***. The unique key ID required for authenticating with Google Cloud. This value is filled in using the "private_key_id" parameter in the JSON credentials file.
   - ***gcp_private_key***. The unique key required for authenticating with Google Cloud. This value is filled in using the "private_key" parameter in the JSON credentials file.
   - ***gcp_client_email***. The generated email associated with the client in the Google Cloud Platform. This value is filled in using the "client_email" parameter in the JSON credentials file.
   - ***gcp_client_id***. The ID associated with the client in the Google Cloud Platform. This value is filled in using the "client_id" parameter in the JSON credential file.
   - ***gcp_client_x509_cert_url***. The client certificate URL associated with the client in the Google Cloud Platform. This value is filled in using the "client_x509_cert_url" parameter in the JSON credential file.
   - ***gcp_pubsub_subscriber_audience***. Type subscriber client URL for your Pub/Sub account.
   - ***gcp_pubsub_publisher_audience***. Type publisher client URL for your Pub/Sub account.
   - ***gcp_pubsub_topic_id***. Type the topic ID for Pub/Sub messages.
   - ***gcp_pubsub_subscription_id***. Type the subscription ID for your Pub/Sub account.
   - ***gcp_pubsub_timeout***. Type the number of seconds between retrying to communicate with Pub/Sub.
   - ***gcp_pubsub_deadline***. Type the number of seconds to retry Pub/Sub after execution. After the defined number of seconds, the SyncPack will stop attempting to communicate with Pub/Sub.
   - ***gcp_pubsub_num_messages***. Type the number of messages to pull from Pub/Sub.
   - ***gcp_task_queue***. Type the name of the queue for the Google Cloud Platform tasks.
   - ***gcp_task_location***. Type the default location for the Google Cloud Platform task.

- ***gcp_task_url***. Type the URL for your Cloud Tasks account.
- ***gcp_task_seconds_from_now***. Type the scheduled time in seconds to create a task in Google Cloud Platform.
- ***gcp_build_steps***. Type the Google Cloud Platform build steps to execute when the build is successful in Cloud Build.
- ***add_template***. Toggle the JSON editor to define the template to translate SL1 event information to a Google Cloud Platform task, error, or message.

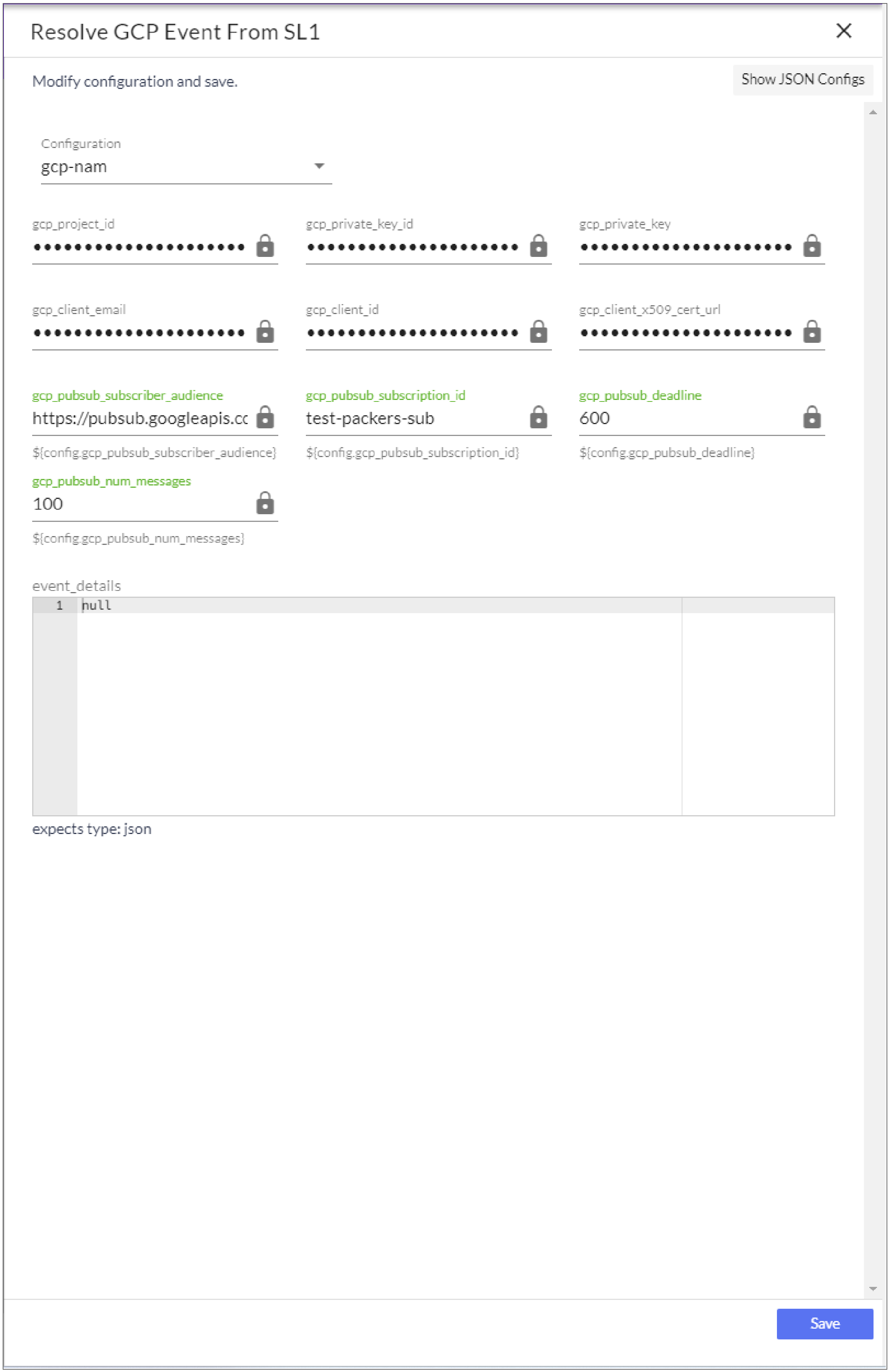
9. Click **[Save]**. You can now align this configuration object with one or more applications.

Aligning a Configuration Object and Configuring PowerFlow Applications

With this SyncPack, any status changes made to an SL1 event is sent to the Google Cloud Platform to update the corresponding task, error, or message. Any status changes to the Google Cloud Platform task or error are synchronized back to the corresponding SL1 event. You will need to align the Google Cloud Platform applications with the relevant configuration object in PowerFlow, and, if needed, update any other fields on the **Configuration** pane for the applications.

To align the configuration object with the relevant PowerFlow applications:

1. On the **Applications** page of the PowerFlow user interface, open one of the PowerFlow applications listed above and click **[Configure]** (⚙). The **Configurations** pane for that application appears:



Resolve GCP Event From SL1

Modify configuration and save.

Configuration
gcp-nam

gcp_project_id: ••••••••••••••••••••••

gcp_private_key_id: ••••••••••••••••••••••

gcp_private_key: ••••••••••••••••••••••

gcp_client_email: ••••••••••••••••••••••

gcp_client_id: ••••••••••••••••••••••

gcp_client_x509_cert_url: ••••••••••••••••••••••

gcp_pubsub_subscriber_audience: https://pubsub.googleapis.cc
${config.gcp_pubsub_subscriber_audience}

gcp_pubsub_subscription_id: test-packers-sub
${config.gcp_pubsub_subscription_id}

gcp_pubsub_deadline: 600
${config.gcp_pubsub_deadline}

gcp_pubsub_num_messages: 100
${config.gcp_pubsub_num_messages}

event_details

```
null
```

expects type: json

2. From the ***Configurations*** drop-down, select the configuration object you want to use.

> **NOTE:** The values for **sl1_hostname** and the other parameters that appear in the **Configuration** pane with a padlock icon (🔒) are populated by the configuration object you aligned with the application. Do not modify these values. If you encounter an error, make sure your configuration object is configured properly.

3. Click **[Save]** to align that configuration with the application.
4. Repeat this process for the other PowerFlow applications.

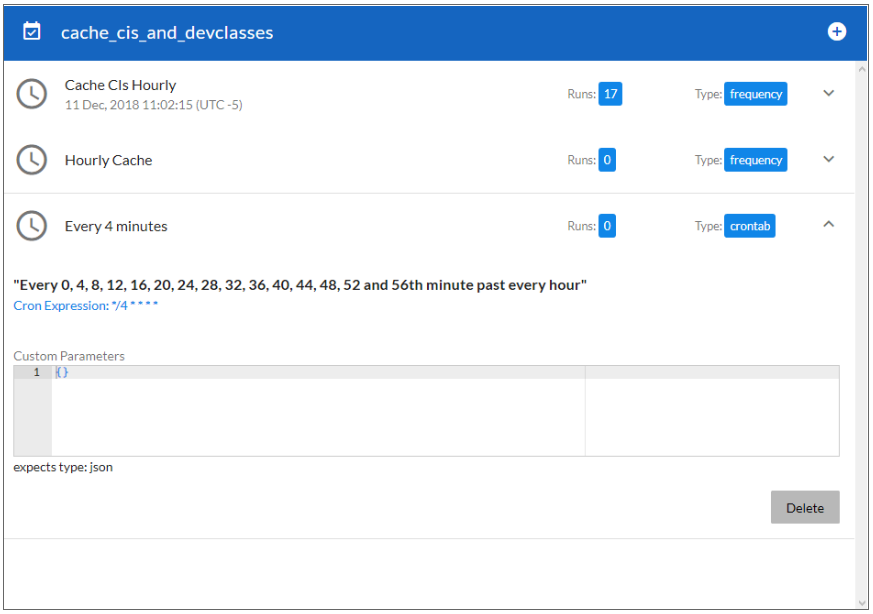
Scheduling PowerFlow Applications

To trigger the applications in the SyncPack, you must schedule the applications included in the SyncPack.

You can create one or more schedules for a single application in the PowerFlow user interface. When creating each schedule, you can specify the queue and the configuration file for that application.

To schedule an application:

1. On the **Applications** page (▦), click the **[Schedule]** button for the application you want to schedule. The **Schedule** window appears, displaying any existing schedules for that application:



cache_cis_and_devclasses

Cache CIs Hourly
11 Dec, 2018 11:02:15 (UTC -5) Runs: 17 Type: frequency

Hourly Cache Runs: 0 Type: frequency

Every 4 minutes Runs: 0 Type: crontab

"Every 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52 and 56th minute past every hour"
Cron Expression: */4 * * * *

Custom Parameters

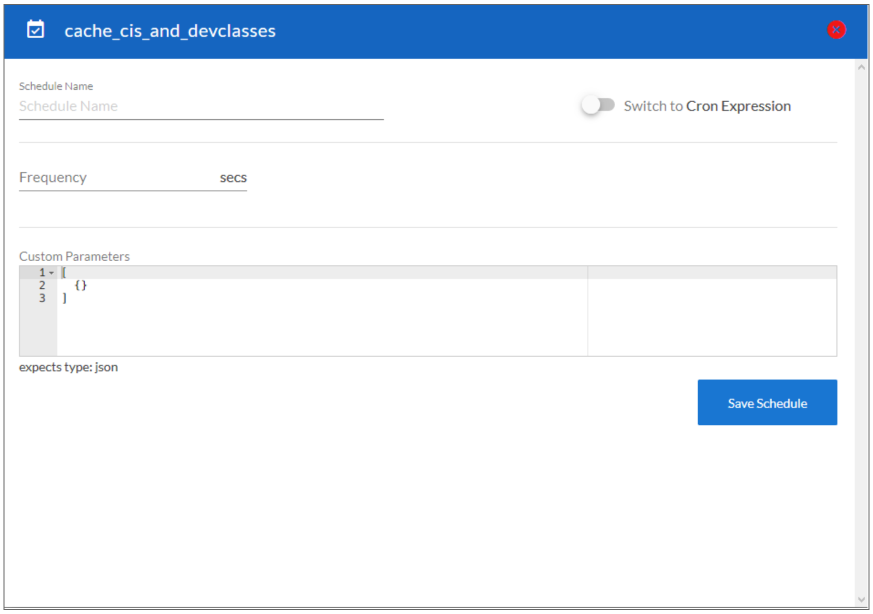
```
{}
```

expects type: json

Delete

NOTE: If you set up a schedule using a cron expression, the details of that schedule display in a more readable format in this list. For example, if you set up a cron expression of */4 * * * *, the schedule on this window includes the cron expression along with an explanation of that expression: "*Every 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, and 56th minute past every hour*".

2. Select a schedule from the list to view the details for that schedule.
3. Click the + icon to create a schedule. A blank **Schedule** window appears:

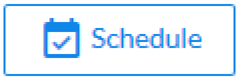


```
[
  {}
]
```

4. In the **Schedule** window, complete the following fields:

   - ***Schedule Name***. Type a name for the schedule.
   - ***Switch to***. Use this toggle to switch between a cron expression and setting the frequency in seconds.
     - ***Cron expression***. Select this option to schedule the application using a cron expression. If you select this option, you can create complicated schedules based on minutes, hours, the day of the month, the month, and the day of the week. As you update the cron expression, the **Schedule** window displays the results of the expression in more readable language, such as *Expression: "Every 0 and 30th minute past every hour on the 1 and 31st of every month*", based on */30 * * /30 * *.
     - ***Frequency in seconds***. Type the number of seconds per interval that you want to run the application.
   - ***Custom Parameters***. Type any JSON parameters you want to use for this schedule, such as information about a configuration file or mappings.

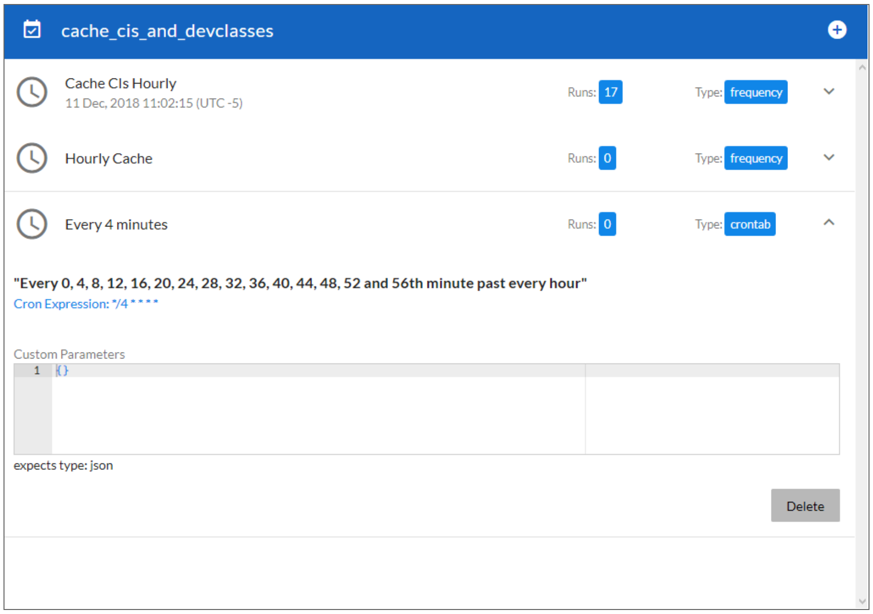
5. Click **[Save Schedule]**. The schedule is added to the list of schedules on the initial **Schedule** window. Also, on the **Applications** page, the word "Scheduled" appears in the **Scheduled** column for this application, and the **[Schedule]** button contains a check mark:



> **NOTE:** After you create a schedule, it continues to run until you delete it. Also, you cannot edit an existing schedule, but you can delete it and create a similar schedule if needed.

To view or delete an existing schedule:

1. On the **Applications** page, click the **[Schedule]** button for the application that contains a schedule you want to delete. The **Schedule** window appears.
2. Click the down arrow icon (✓) to view the details of an existing schedule:



cache_cis_and_devclasses

Cache CIs Hourly
11 Dec, 2018 11:02:15 (UTC -5)
Runs: 17
Type: frequency

Hourly Cache
Runs: 0
Type: frequency

Every 4 minutes
Runs: 0
Type: crontab

"Every 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52 and 56th minute past every hour"
Cron Expression: */4 * * * *

Custom Parameters

```
{}
```

expects type: json

3. To delete the selected schedule, click **[Delete]**. The schedule is removed.

> **NOTE:** When either multiple SL1 instances or multiple Google Cloud Platform instances are involved with PowerFlow, you should create an individual configuration object for each SL1 or Google Cloud Platform instance. Next, create an individual schedule for each configuration object. Each schedule should use a configuration object that is specific to that single SL1 or Google Cloud instance. Creating copies of a PowerFlow application from a SyncPack for the purpose of distinguishing between domains is not supported, and will result in issues on upgrades.

Chapter

3

Introduction to the Google Cloud Automation PowerPack

Overview

This chapter describes how to install the *Google Cloud Automation* PowerPack.

This chapter covers the following topics:

What is the Google Cloud Automation PowerPack?

The *Google Cloud Automation* PowerPack contains Run Book automation and action policies that you can use to integrate with the *Google Cloud Platform* SyncPack.

Installing the Google Cloud Automation PowerPack

Before completing the steps in this manual, you must import and install the latest version of the *Google Cloud Automation* PowerPack.

NOTE: The *Google Cloud Automation* PowerPack requires SL1 version 11.1.0 or later. For details on upgrading SL1, see the appropriate SL1 Release Notes.

TIP: By default, installing a new version of a PowerPack overwrites all content from a previous version of that PowerPack that has already been installed on the target system. You can use the ***Enable Selective PowerPack Field Protection*** setting in the **Behavior Settings** page (System > Settings > Behavior) to prevent new PowerPacks from overwriting local changes for some commonly customized fields. (For more information, see the ***System Administration*** manual.)

To download and install a PowerPack:

1. Download the PowerPack from the ScienceLogic Support Site at https://support.sciencelogic.com/s/powerpacks.
2. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
3. In the **PowerPack Manager** page, click the **[Actions]** button, then select *Import PowerPack*. The **Import PowerPack** dialog box appears.
4. Click the **[Browse]** button and navigate to the PowerPack file.
5. When the **PowerPack Installer** modal appears, click the **[Install]** button to install the PowerPack.

NOTE: If you exit the **PowerPack Installer** modal without installing the imported PowerPack, the imported PowerPack will not appear in the **PowerPack Manager** page. However, the imported PowerPack will appear in the **Imported PowerPacks** modal. This page appears when you click the **[Actions]** menu and select *Install PowerPack*.

Chapter

4

Configuring Automation Action Credentials

Overview

This chapter describes how to configure the credential required by the automation actions in the *Google Cloud Automation* PowerPack.

This chapter covers the following topics:

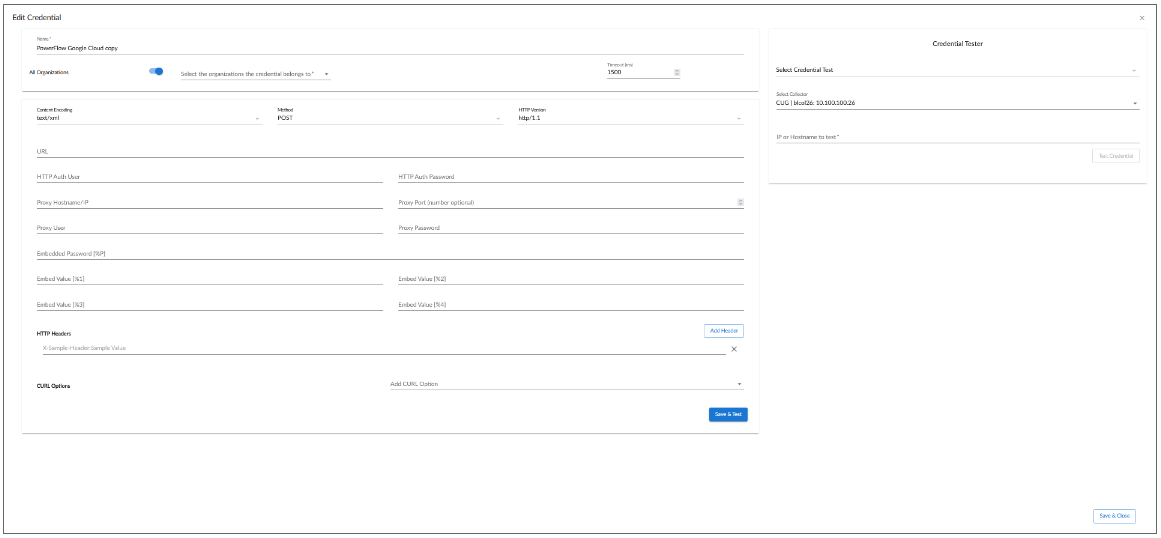
Creating a SOAP/XML Credential for Google Cloud Platform

To configure SL1 to monitor a Google Cloud Platform instance, you must create a SOAP/XML credential to enable the Dynamic Applications in the PowerPack to communicate with the Google Cloud Platform and PowerFlow. The *Google Cloud Automation* PowerPack includes a template for a SOAP/XML credential that you can edit for use with your PowerFlow system.

NOTE: If you are on an SL1 system prior to version 11.1.0, you will not be able to duplicate the sample credential. It is recommended that you create your new credentials using ***the SL1 classic user interface*** so you do not overwrite the sample credential(s).

To create a SOAP/XML credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the "PowerFlow Google Cloud" sample credential, click its **[Actions]** icon (⋯), and then select *Duplicate*. A copy of the credential, called **PowerFlow Google Cloud copy** appears.



Edit Credential

Name*: PowerFlow Google Cloud copy

All Organizations

Timeout (ms): 1500

Content Encoding: text/xml

Method: POST

HTTP Version: http/1.1

Credential Tester

Select Credential Test

Select Collector: CUG | blcol26: 10.100.100.26

3. Supply values in the following fields:

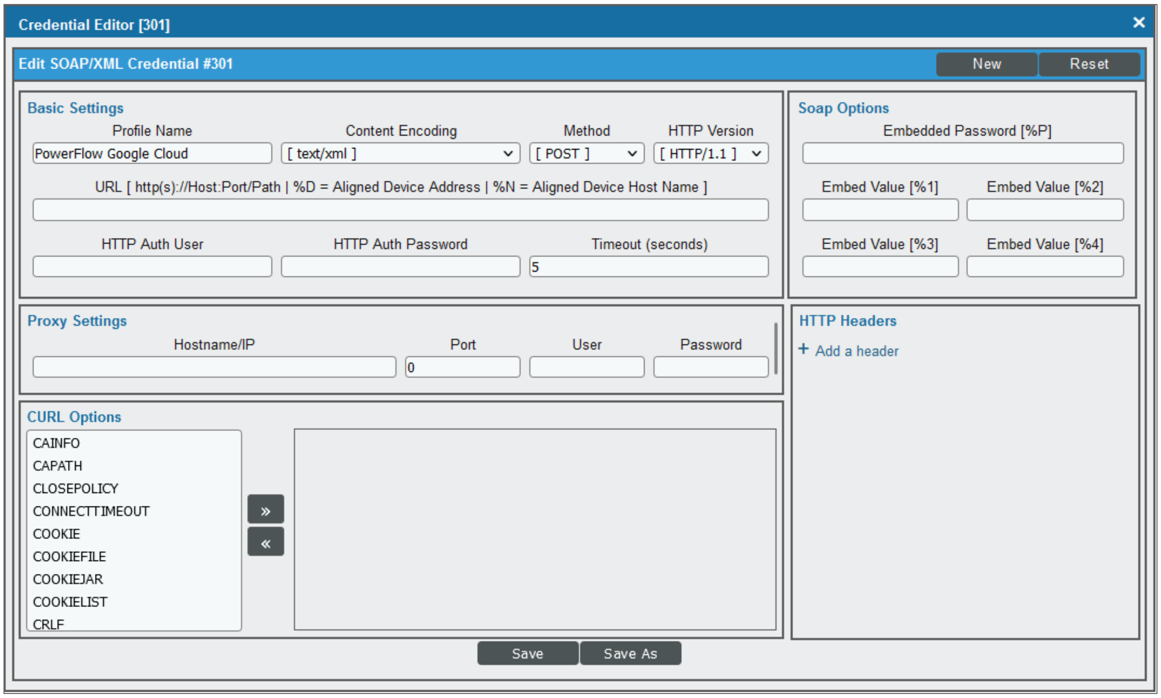
    - ***Name***. Type a new name for the credential.
    - ***All Organizations***. Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the ***What organization manages this service?*** drop-down field to align the credential with those specific organizations.
    - ***URL***. Type the URL for your PowerFlow system.
    - ***HTTP Auth User***. Type the username for your PowerFlow system.
    - ***HTTP Auth Password***. Type the password for your PowerFlow system.

4. Click **[Save & Close]**.
5. SL1 assigns the credential an ID number. Take note of the ID number for the new credential on the **Credentials** page, in the **ID** column. You will need the ID number when editing the input parameters of the automation actions included in the *Google Cloud Automation* PowerPack.

Creating a SOAP/XML Credential to Access SL1 PowerFlow in the Classic User Interface

To define a SOAP/XML credential using the example credential:

2. Go to the **Credential Management** page (System > Manage > Credentials).
3. Click the wrench icon () for the **PowerFlow Google Cloud** credential. The **Credential Editor** modal window appears:



Credential Editor [301]

Edit SOAP/XML Credential #301

Profile Name: PowerFlow Google Cloud

Content Encoding: [text/xml]

Method: [POST]

HTTP Version: [HTTP/1.1]

Timeout (seconds): 5

Port: 0

4. Supply values in the following fields:
   - ***Profile Name***. Type a new name for the credential.
   - ***URL***. Type the URL for your SL1 PowerFlow system.
   - ***HTTP Auth User***. Type the username for your SL1 PowerFlow user account.
   - ***HTTP Auth Password***. Type the password for your SL1 PowerFlow user account.
5. Click the **[Save As]** button to save the new SOAP/XML credential.
6. SL1 assigns the credential an ID number. Take note of the ID number that appears in the **Credential Editor** heading, as you will need this when editing the input parameters of the automation actions included in the *Google Cloud Automation* PowerPack.

Chapter

5

Configuring the Google Cloud Automation Run Book Action Policies

Overview

This chapter describes how to edit the Run Book Action policies included in the *Google Cloud Automation* PowerPack so that the action policies can communicate with your SL1 PowerFlow system.

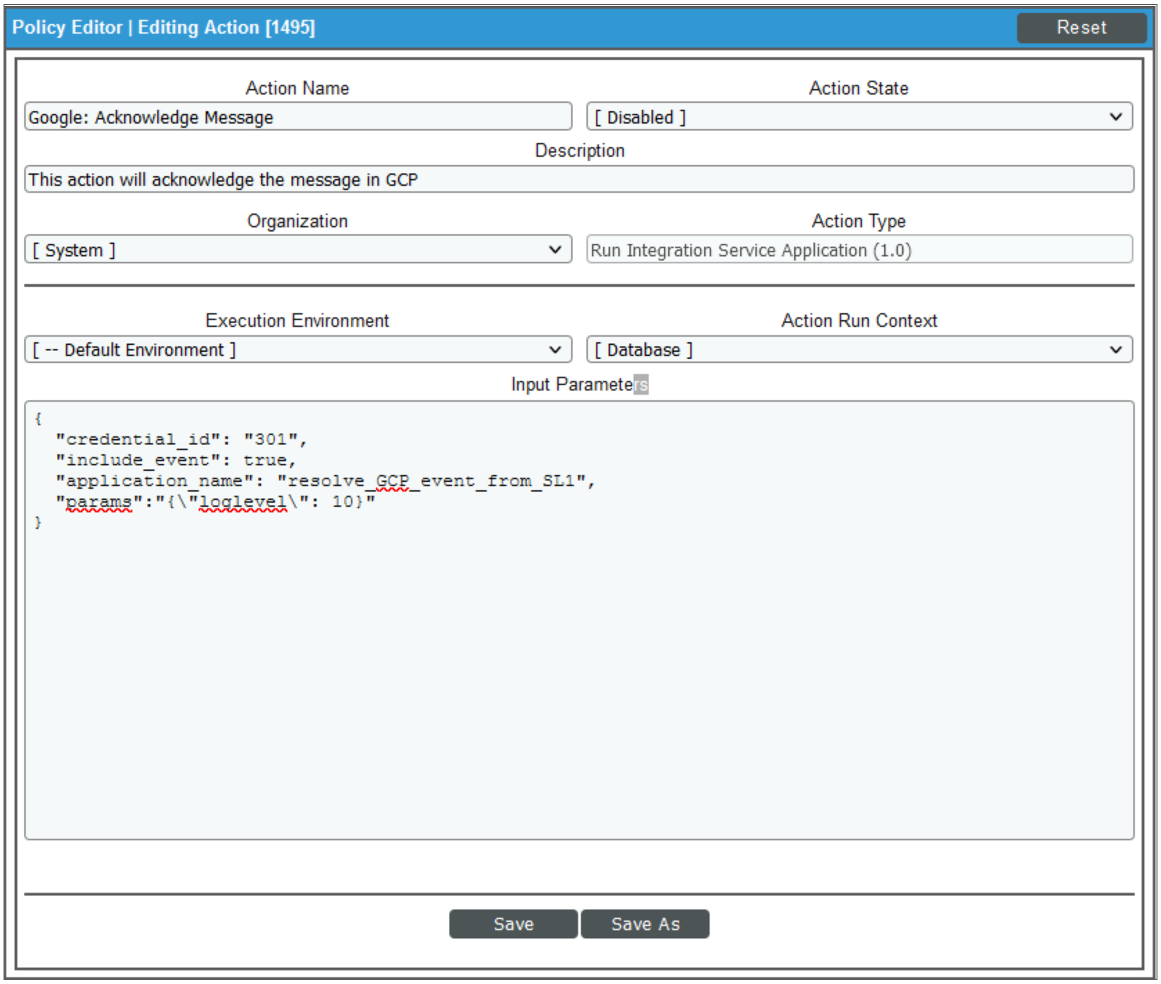
This chapter covers the following topics:

Editing the Google Cloud Automation Action Policies

The *Google Cloud Automation* PowerPack includes six action policies that use the "Run Integration Service Application" action type to trigger the PowerFlow application that sends and receives data to the Google Cloud Platform. You can specify the credential ID in a JSON structure that you enter in the ***Input Parameters*** field in the **Action Policy Editor** modal.

To edit the action policies included in the PowerPack:

1. Go to the **Action Policy Manager** page (Registry > Run Book > Actions).
2. Locate the action policy that you want to use, and then click its wrench icon (🔧). The **Editing Action** page appears:



Action Name: Google: Acknowledge Message

Action State: [Disabled]

Description: This action will acknowledge the message in GCP

Organization: [System]

Action Type: Run Integration Service Application (1.0)

Execution Environment: [-- Default Environment]

Action Run Context: [Database]

Input Parameters

```
{
  "credential_id": "301",
  "include_event": true,
  "application_name": "resolve_GCP_event_from_SL1",
  "params":"{\"loglevel\": 10}"
}
```

3. Select *Enabled* from the ***Action State*** field if it is not already selected.
4. In the ***Input Parameters*** field, change the values of the following parameters:
   - ***credential_id***. Change the value to the credential ID that you noted earlier when *creating a credential for your PowerFlow system*. This parameter is required.

- ***include_event***. Leave the value as "true".
- ***application_name***. Leave the default application value.
- ***params***. Leave the default parameter value.

4. Click **[Save]**.

Chapter

6

Google Cloud Automation Run Book Automation Policies

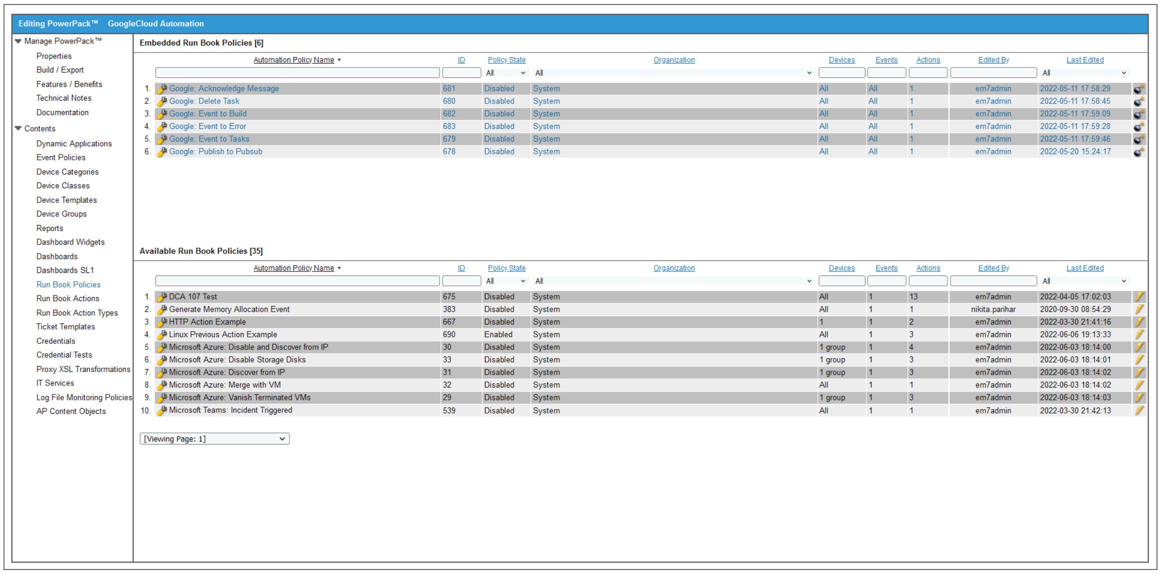
Overview

This chapter describes the Run Book Automation policies found in the *Google Cloud Automation* PowerPack.

This chapter covers the following topics:

Standard Automation Policies

The *Google Cloud Automation* PowerPack includes six standard automation policies that you can enable, shown in the following figure.



Editing PowerPack™ GoogleCloud Automation

- Manage PowerPack™
  - Properties
  - Build / Export
  - Features / Benefits
  - Technical Notes
  - Documentation
- Contents
  - Dynamic Applications
  - Event Policies
  - Device Categories
  - Device Classes
  - Device Templates
  - Device Groups
  - Reports
  - Dashboard Widgets
  - Dashboards
  - Dashboards SL1
  - Run Book Policies
  - Run Book Actions
  - Run Book Action Types
  - Ticket Templates
  - Credentials
  - Credential Tests
  - Proxy XSL Transformations
  - IT Services
  - Log File Monitoring Policies
  - AP Content Objects

Embedded Run Book Policies [6]

	Automation Policy Name	ID	Policy State	Organization	Devices	Events	Actions	Edited By	Last Edited
			All	All					All
1.	Google: Acknowledge Message	681	Disabled	System	All	All	1	em7admin	2022-05-11 17:58:29
2.	Google: Delete Task	680	Disabled	System	All	All	1	em7admin	2022-05-11 17:58:45
3.	Google: Event to Build	682	Disabled	System	All	All	1	em7admin	2022-05-11 17:59:09
4.	Google: Event to Error	683	Disabled	System	All	All	1	em7admin	2022-05-11 17:59:28
5.	Google: Event to Tasks	679	Disabled	System	All	All	1	em7admin	2022-05-11 17:59:46
6.	Google: Publish to Pubsub	678	Disabled	System	All	All	1	em7admin	2022-05-20 15:24:17

Available Run Book Policies [35]

	Automation Policy Name	ID	Policy State	Organization	Devices	Events	Actions	Edited By	Last Edited
			All	All					All
1.	DCA 107 Test	675	Disabled	System	All	1	13	em7admin	2022-04-05 17:02:03
2.	Generate Memory Allocation Event	383	Disabled	System	All	1	1	nikita.parihar	2020-09-30 08:54:29
3.	HTTP Action Example	667	Disabled	System	1	1	2	em7admin	2022-03-30 21:41:16
4.	Linux Previous Action Example	690	Enabled	System	All	1	3	em7admin	2022-06-06 19:13:33
5.	Microsoft Azure: Disable and Discover from IP	30	Disabled	System	1 group	1	4	em7admin	2022-06-03 18:14:00
6.	Microsoft Azure: Disable Storage Disks	33	Disabled	System	1 group	1	3	em7admin	2022-06-03 18:14:01
7.	Microsoft Azure: Discover from IP	31	Disabled	System	1 group	1	3	em7admin	2022-06-03 18:14:02
8.	Microsoft Azure: Merge with VM	32	Disabled	System	All	1	1	em7admin	2022-06-03 18:14:02
9.	Microsoft Azure: Vanish Terminated VMs	29	Disabled	System	1 group	1	3	em7admin	2022-06-03 18:14:03
10.	Microsoft Teams: Incident Triggered	539	Disabled	System	All	1	1	em7admin	2022-03-30 21:42:13

These policies synchronize events that occur in SL1 and automate creating tasks, deleting tasks, starting builds, reporting errors, and publishing Pub/Sub API messages with Google Cloud Platform.

When an event is detected in SL1, a task, build, or error is triggered in the Google Cloud Platform. When a message is detected in SL1, the associated message is acknowledged in the Google Cloud Platform and published to Pub/Sub.

The following table shows the automation policy, its default aligned events, and the automation action that runs in response to the events.

Automation Policy Name	Aligned Events	Automation Action
Google: Acknowledge Message	All events	Google: Acknowledge Message
Google: Delete Task	All events	Google: Delete Task
Google: Event to Build	All events	Google: Event to Build
Google: Event to Error	All events	Google: Event to Error
Google: Event to Task	All events	Google: Event to Task
Google: Publish to Pubsub	All events	Google: Publish to Pubsub

If a policy is not already enabled, you can enable it by doing the following:

1. In SL1, go to the **Automation Policy Manager** page (Registry > Run Book > Automation).
2. Locate the automation policy you want to enable and click its wrench icon (🔧). The **Automation Policy Editor** page appears.
3. Update the following fields:
   - ***Policy State***. Select *Enabled*.
   - ***Policy Priority***. Select *Default* to ensure that this PowerFlow automation policy is added to the top of the queue.
   - ***Available Actions***. If it is not already selected, select the corresponding Google Cloud Run Book action.

By default, the "GCP: Publish to Pubsub" automation policy will create Google Cloud Platform Events for all devices. You can limit the devices affected by making changes to the ***Organization***, ***Severity***, ***Match Logic***, ***Aligned Devices***, and/or ***Aligned Events*** fields.

> NOTE: ScienceLogic does not recommend making changes to the ***Policy Type***, ***Repeat Time***, or ***Align With*** fields or changing the ***Criteria Logic*** to *and event is NOT acknowledged*.

4. Click **[Save]**.
5. Repeat steps for any other Run Book Automation policies that need to be enabled.

Chapter 7

Troubleshooting

Overview

This chapter contains troubleshooting resources and procedures to use with the *Google Cloud Platform* SyncPack.

This chapter covers the following topics:

Initial Troubleshooting Steps

The first step taken when troubleshooting should be to ensure that there are no issues with the data platforms with which PowerFlow is interacting. There might be additional configurations or actions enabled on Google Cloud or SL1 that result in unexpected behavior. Perform the steps in the following sections to troubleshoot SL1 PowerFlow and Google Cloud. For detailed information about how to perform the steps below, see ***Resources for Troubleshooting***.

SL1 PowerFlow

1. Run `docker service ls` on the PowerFlow server.
2. Note the Docker container version, and verify that the Docker services are running.
3. If a certain service is failing, make a note the service name and version.
4. Run `docker service ps <service_name>` to see the historical state of the failing service and make a note of this information. For example: `docker service ps iservices_contentapi`
5. Make a note of any logs impacting the service by running `docker service logs <service_name>`. For example: `docker service logs iservices_couchbase`

Google Cloud Platform

1. Make a note of the SyncPack version, if applicable.
2. Make a note of the Google Cloud application that is failing on PowerFlow.
3. Make a note of what step is failing in the application. Try running the application in debug mode, and capture any traceback or error messages that occur in the step log.

Resources for Troubleshooting

This section contains port information for PowerFlow and troubleshooting commands for Docker, Couchbase, and the PowerFlow API.

Useful PowerFlow Ports

- **https://*<IP of PowerFlow>*:8091**. Provides access to Couchbase, a NoSQL database for storage and data retrieval.
- **https://*<IP of PowerFlow>*:15672**. Provides access to the RabbitMQ Dashboard, which you can use to monitor the service that distributes tasks to be executed by PowerFlow workers. Use the following for login: guest guest for the login.
  - Username: guest
  - Password: guest
- **https://*<IP of PowerFlow>*/flower**. Provides access to Flower, a tool for monitoring and administrating Celery clusters.

NOTE: For version 2.0.0 and later of PowerFlow, port 5556 must be open for both PowerFlow and the client.

Helpful Docker Commands

PowerFlow is a set of services that are containerized using Docker. For more information about Docker, see the Docker tutorial.

Use the following Docker commands for troubleshooting and diagnosing issues with PowerFlow:

Viewing Container Versions and Status

To view the PowerFlow version, SSH in to your PowerFlow instance and run the following command:

```
docker service ls
```

In the results, you can see the container ID, name, mode, status (see the replicas column), and version (see the image column) for all the services that make up PowerFlow.

Restarting a Service

Run the following command to restart a single service:

```
docker service update --force <service_name>
```

Stopping all PowerFlow Services

Run the following command to stop all PowerFlow services:

```
docker stack rm iservices
```

Restarting Docker

Run the following command to restart Docker:

```
systemctl restart docker
```

NOTE: Restarting Docker does not clear the queue.

Viewing Logs for a Specific Service

You can use the Docker command line to view the logs of any currently-running service in the PowerFlow cluster. To view the logs of a service, run the following command:

```
docker service logs -f iservices_<service_name>
```

Some common examples include the following:

```
docker service logs –f iservices_couchbase
```

```
docker service logs –f iservices_steprunner
```

```
docker service logs –f iservices_contentapi
```

Application logs are stored on the central database as well as on all Docker hosts in a clustered environment. These logs are stored at `/var/log/iservices` for both single-node or clustered environments. However, the logs on each Docker host relate only to the services running on that host. For this reason, using the Docker service logs is the best way to get logs from all hosts at once.

Clearing RabbitMQ Volume

RabbitMQ is a service that distributes tasks to be executed by PowerFlow workers.

The following error message might appear if you try to run a PowerFlow application via the API:

```
Internal error occurred: Traceback (most recent call last):\n File
\"./content_api.py\", line 199, in kickoff_application\n task_status =
... line 623, in _on_close\n (class_id, method_id),
ConnectionError)\nInternalError: Connection.open: (541) INTERNAL_ERROR -
access to vhost '/' refused for user 'guest': vhost '/' is down
```

First, verify that your services are up. If there is an issue with your RabbitMQ volume, you can clear the volume with the following commands:

```
docker service rm iservices_rabbitmq
```

```
docker volume rm iservices_rabbitdb
```

If you get a message stating that the volume is in use, run the following command:

```
docker rm <id of container using volume>
```

Re-deploy PowerFlow by running the following command:

```
docker stack deploy -c /opt/iservices/scripts/docker-compose.yml
iservices
```

NOTE: Restarting Docker does not clear the queue, because the queue is persistent. However, clearing the queue with the commands above might result in data loss due to the tasks being removed from the queue.

Viewing the Process Status of All Services

Run the following command:

```
docker ps
```

Deploying Services from a Defined Docker Compose File

Run the following command:

```
docker stack deploy -c <compose-file> iservices
```

Dynamically Scaling for More Workers

Run the following command:

```
docker service scale iservices_steprunner=10
```

Completely Removing Services from Running

Run the following command:

```
docker stack rm iservices
```

Diagnostic Tools

Multiple diagnostic tools exist to assist in troubleshooting issues with the PowerFlow platform:

- ***Docker PowerPack***. This PowerPack monitors your Linux-based PowerFlow server with SSH (the PowerFlow ISO is built on top of an Oracle Linux Operating System). This PowerPack provides key performance indicators about how your PowerFlow server is performing. For more information on the Docker PowerPack and other PowerPacks that you can use to monitor PowerFlow, see the *Using SL1 to Monitor SL1 PowerFlow* chapter in the ***SL1 PowerFlow Platform*** manual.
- ***Flower***. This web interface tool can be found at the `/flower` endpoint. It provides a dashboard displaying the number of tasks in various states as well as an overview of the state of each worker. This tool shows the current number of active, processed, failed, succeeded, and retried tasks on the PowerFlow platform. This tool also shows detailed information about each of the tasks that have been executed on the platform. This data includes the UUID, the state, the arguments that were passed to it, as well as the worker and the time of execution. Flower also provides a performance chart that shows the number of tasks running on each individual worker.
- ***Debug Mode***. All applications can be run in "debug" mode via the PowerFlow API. Running applications in debug mode may slow down the platform, but they will result in much more detailed logging information that is helpful for troubleshooting issues. For more information on running applications in Debug Mode, see ***Retrieving Additional Debug Information***.

- ***Application Logs***. All applications generate a log file specific to that application. These log files can be found at `/var/log/iservices` and each log file will match the ID of the application. These log files combine all the log messages of all previous runs of an application up to a certain point. These log files roll over and will get auto-cleared after a certain point.
- ***Step Logs***. Step logs display the log output for a specific step in the application. These step logs can be accessed via the PowerFlow user interface by clicking on a step in an application and bringing up the **Step Log** tab. These step logs display just the log output for the latest run of that step.
- ***Service Logs***. Each Docker service has its own log. These can be accessed via SSH by running the following command:

```
docker service logs -f <service_name>
```

Retrieving Additional Debug Information (Debug Mode)

The logs in PowerFlowuse the following **loglevel** settings, from most verbose to least verbose:

- ***10***. Debug Mode.
- ***20***. Informational.
- ***30***. Warning. This is the default settings if you do not specify a loglevel.
- ***40***. Error.

WARNING: If you run applications with "loglevel": 10, those applications will take longer to run because of increased I/O requirements. Enabling debug logging using the following process is the only recommended method. **ScienceLogic does not recommend setting "loglevel": 10 for the whole stack with the docker-compose file.**

To run an application in Debug Mode using the PowerFlow user interface:

1. Select the PowerFlow application from the **Applications** page.
2. Hover over the **Run** (▶) from and select ***Debug Run*** from the pop-up menu. PowerFlow executes the application in Debug Mode with a log level of 10.

To run an application in Debug Mode using the API:

1. POST the following to the API endpoint:

```
https://<PowerFlow>/api/v1/applications/run
```

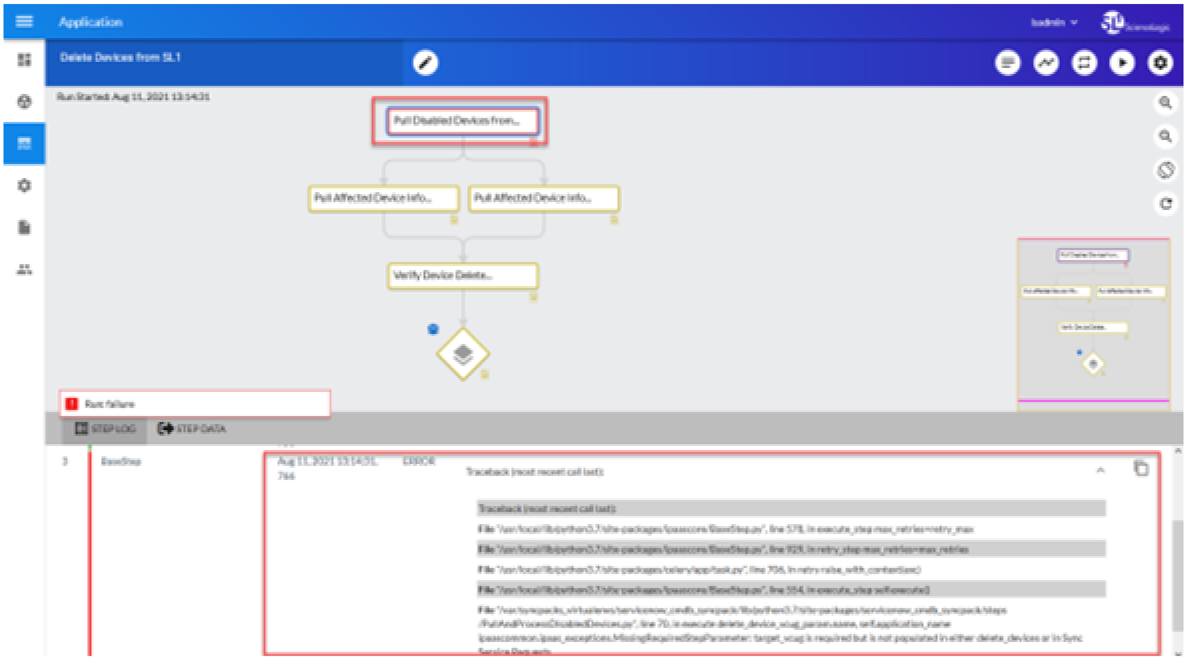
2. Include the following in the request body:

```
{
  "name": "<application_name>",
  "params": {
        "loglevel": 10

    }

}
```

After running the application in Debug Mode, review the step logs in the PowerFlow user interface to see detailed debug output for each step in the application. This information is especially helpful when trying to understand why an application or step failed:


Application
isadmin
Delete Devices from SL1
Run Started: Aug 11, 2021 13:14:31
Pull Disabled Devices from...
Pull Affected Device Info...
Pull Affected Device Info...
Verify Device Delete...
Run failure
STEP LOG
STEP DATA
3
BaseStep

```
Aug 11, 2021 13:14:31.764   ERROR   Traceback (most recent call last):

Traceback (most recent call last):
File "/usr/local/lib/python3.7/site-packages/ipaascore/BaseStep.py", line 578, in execute_step max_retries=retry_max
File "/usr/local/lib/python3.7/site-packages/ipaascore/BaseStep.py", line 929, in retry_step max_retries=max_retries
File "/usr/local/lib/python3.7/site-packages/celery/app/task.py", line 706, in retry raise_with_context(exc)
File "/usr/local/lib/python3.7/site-packages/ipaascore/BaseStep.py", line 554, in execute_step self.execute()
File "/var/syncpacks_virtualenvs/servicenow_cmdb_syncpack/lib/python3.7/site-packages/servicenow_cmdb_syncpack/steps
/PullAndProcessDisabledDevices.py", line 70, in execute delete_device_vcug_param.name, self.application_name
ipaascommon.ipaas_exceptions.MissingRequiredStepParameter: target_vcug is required but is not populated in either delete_devices or in Sync
Service Requests
```

You can also run an application in debug using curl via SSH:

1. SSH in to the PowerFlowinstance.

2. Run the following command:

```
curl -v -k -u isadmin:em7admin -X POST "https://<your_
hostname>/api/v1/applications/run" -H 'Content-Type: application/json' -
H 'cache-control: no-cache' -d '{"name": "interface_sync_sciencelogic_
to_servicenow","params": {"loglevel": 10}}'
```

LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described in this Site at any time without notice.

Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: legal@sciencelogic.com. For more information, see https://sciencelogic.com/company/legal.

ScienceLogic

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010