Configuring the Zebrium Connector for SL1

The Zebrium Connector, also called the **ze_connector** service, continually checks your Zebrium instance for suggestions and alerts. The Connector then looks for an SL1 device that match the Zebrium alerts, and sends the Zebrium suggestions and alerts to that device in SL1.

As a result, the Zebrium Connector lets you view Zebrium suggestions and alerts in the following locations in SL1:

- The **Events** page
- The **Event Investigator** page for a Zebrium suggestion or alert
- The **[Investigator]** tab and the **[Events]** tab of the **Device Investigator** page
- The **Timeline** widget and the **[Log Insights]** tab of the **Service Investigator** page

Workflow for Configuring the Connector

Before you can view Zebrium data on these SL1 pages, you will need to complete the following configuration steps in Zebrium and SL1:

- Configure Zebrium:
  - *Create an authentication token in Zebrium*
- Configure SL1:
  - *Create a service connection in SL1*
  - *Create an SL1 authentication token*
  - *Create a default virtual device* (optional)
  - *Install the Zebrium Event Policies PowerPack*
- Configure the Zebrium Connector:
  - *Download and install the RPM file for the Zebrium Connector*
  - *Configure the config.yaml file*

Creating an Authentication Token in Zebrium

You first need to access the Zebrium user interface to get an authorization token, which you will use in the SL1 setup.

To create an authorization token in Zebrium:

1. In the Zebrium user interface, go to the **Access Tokens** page (Settings () > Access Tokens).
2. Click **[Add Access Token]**. The **Add Access Token** dialog appears.

3. Type a ***Name*** and select a ***Role*** of *viewer*.
4. Select the ***Deployment*** you want to monitor.
5. Click **[Add]**. The new token is added to the **Access Tokens** page. The token is in the format "Bearer <token>", such as *Bearer abcdefghijk*.
6. Hover over the **Name/Token** column of the new token and click the **[Copy]** button that appears.
7. Save the access token for the next set of steps.

Configuring SL1

Complete the following steps to configure SL1 so it can use the Zebrium Connector.

Create a Service Connection in SL1

To create a service connection in SL1:

1. In SL1, go to the **Service Connections** page (Manage > Service Connections).
2. Click **Add Service Connection**. The **Create Zebrium Connection** window appears.
3. Complete the following fields:
    - ***Name***. Type a name for this new service connection.
    - ***Access Token***. Paste the access token you created in Zebrium into this field. You can view this information on the **Access Tokens** page (Settings ( ) > Access Tokens) in the Zebrium user interface.
    - ***Zebrium Endpoint URL***. Add the endpoint URL for your Zebrium instance. Zebrium Cloud users can use the default value in this field, while Zebrium On Prem users will need to add the URL of their on-premises Zebrium instance.
    - ***Share data with***. Select the *All Organizations* toggle (turn it blue) to share this connection with all existing and newly created organizations. Alternately, you deselect the *All Organizations* toggle (turn it gray) and select one or more organizations from the ***Selected Organizations*** drop-down to limit access to this connection to only those organizations.
4. Click **[Save]**. The service connection is added to the **Service Connections** page.

Create an SL1 Authentication Token

Next, you will need to encode your SL1 credentials to create an SL1 authentication token:

1. Go to a Base64 encoding site like https://www.base64encode.org and paste your SL1 username and password in the text box. Use the following format:

    `<username>:<password>`

    For example: `myuser:mypassword`

2. Use the default settings and click **[Encode]**. Your encoded credentials will look like the following:

   ```
   bXl1c2VyOm15cGFzc3dvcmQ=
   ```

   > **NOTE:** The authentication token is in the format "Basic <token>".

3. Copy the newly encoded credentials, which will work as your SL1 authentication token.

Create a Default Virtual Device (optional)

The Zebrium Connector can send Zebrium suggestions and alerts to any device in SL1. If you do not have a specific device that you want to use for this purpose, you can optionally configure a "default" SL1 device. The Connector will send any Zebrium suggestions and alerts that do not map to existing SL1 devices to this default device.

For this purpose, you can create a virtual device in SL1 to receive all of these unassigned suggestions and alerts.

To create a default virtual device in SL1:

1. Ensure that SL1 includes a device class for virtual devices. These device classes must have a device category of "virtual" and a collection type of "virtual".
2. On the **Device Manager** page (Devices > Device Manager), click the **[Actions]** button and select *Create Virtual Device*. The **Create Virtual Device** modal appears.
3. Complete the following fields:
   - ***Device Name***. Name of the virtual device.
   - ***Organization***. Organization to associate with the virtual device. Select from the drop-down list of all organizations in SL1.
   - ***Device Class***. The device class to associate with the virtual device. Select from the drop-down list of device classes. Only device classes with a device category of "virtual" and a collection type of "virtual" appear in the list.
   - ***Collector***. Specifies which instance of SL1 will perform auto-discovery and gather data from the device. Can also specify a "virtual" connector. Select from the drop-down list of all collectors in SL1.
4. Click **[Add]** to save the new virtual device. SL1 displays the new device ID after the text **Device Added**.
5. Before you close the modal, make a note of the ID for your new virtual device. You can sort for this ID on the **Devices** page in SL1 to quickly locate this new virtual device.

Install the Zebrium Event Policies PowerPack

To convert the API alerts sent by the Zebrium Connector into SL1 events, you will need the Zebrium event policies, which are available in the "Zebrium Event Policies" PowerPack. The event policies will be automatically enabled when you install the PowerPack.

To configure the Zebrium event policies:

1. Download and install the "Zebrium Event Policies" PowerPack. For more information, see Importing and Installing a PowerPack.
2. Go to the **Event Policies** page (Events > Event Policies) and sort by "Zebrium" in the **Name** column.
3. Make sure all of the Zebrium event policies from the PowerPack have a **Status** of *Enabled*. If not, check the boxes for the policies that are not enabled and click **[Enable]**.

Configuring the Zebrium Connector

The Zebrium Connector, also called the **ze_connector** service, continually checks your Zebrium instance for suggestions and alerts. The Connector then looks for an SL1 device that match the Zebrium alerts, and sends the Zebrium suggestions and alerts to that device in SL1.

You will need to install the Zebrium Connector RPM file on the SL1 server that you want to connect with Zebrium.

Complete the following steps to configure the Zebrium Connector (ze_connector).

Prerequisites

The SL1 server where you install this service must have the following:

- systemd
- Python 3.8
- `sudo` access to the server
- SL1 version 12.2.0 or later, running Oracle Linux 8 or later, with the "Zebrium Event Policies" PowerPack installed

IMPORTANT: ScienceLogic strongly recommends that you create a separate SL1 account for the Zebrium integration instead of using the default "em7admin" user account. For more information, see *Manually Creating a New User Account*.

Download and Install the RPM file for the Connector

You will need to download the RPM file from the ScienceLogic Support site, and then upload it to your SL1 system.

To download and install the RPM file:

1. Go to the ScienceLogic Support site at https://support.sciencelogic.com/s/.
2. Click the **[Product Downloads]** tab and select *SL1 Platform*. The **Platform Downloads** page appears.
3. Click the link for **SL1 Hollywood Platform 12.2**. The **Release Version** page appears.
4. In the **Release Files** section, click the RPM link for the **Zebrium Ze_Connector** RPM file. A **Release File** page appears.
5. Click **[Download File]** at the bottom of the **Release File** page.

6. SSH to the server where you are installing the RPM. and run the following command to install the RPM:
```
sudo dnf install ze_connector-0.0.2-1.el8.noarch.rpm -y
```

7. *Configure the config.yaml file as needed:*
```
sudo vi /usr/bin/ze_connector/config.yaml
```

8. Restart the service and verify:
```
sudo systemctl restart zeconnector

sudo systemctl status zeconnector

sudo journalctl -u zeconnector

tail /usr/bin/ze_connector/out.log
```

Configure the config.yaml file

The **/usr/bin/ze_connector/config.yaml** file is supplied as part of the RPM install. You can use this sample configuration file to set up new jobs. This section explains the structure of the **config.yaml** file. You can copy this file and update it for the connector jobs.

NOTE: This schema will be overwritten to track the most recent Zebrium event found, specifically the `poll_timing.poll_start_time_iso` field.

Configuration Schema

- `jobs`: (array, required) - polling jobs to run
    - `name`: (str, required) - unique name of this job for log message readability
    - `sl1_api_config`: (obj, required)
        - `api_url`: (str, required) - URL endpoint for the SL1 API to query
        - `api_auth`: (str, required) - Basic auth token for the SL1 API (see *Create an SL1 Authentication Token* for format)
    - `poll_timing`: (obj, optional)
        - `poll_sleep_seconds`: (int, optional default:60) - number of seconds to sleep between polling requests
        - `poll_start_time_iso`: (str, optional default:now) - ISO 8601 timestamp for when to start querying for Zebrium alerts

- `sl1_default_device_ids`: (array[str], optional default:[]) - list of SL1 device IDs to send alerts to if no device is matched automatically; omit to not send an alert if no device is matched
- `ze_deployment_id`: (str, required) - name of the Zebrium deployment to query
- `ze_service_groups`: (array[str], optional default:[]) - list of Zebrium service groups to query
- `sl1_override_event_time`: (bool, optional default:False) - overrides using the Zebrium alert timestamp and instead uses now as when the alert occurred

Example Configuration

The following configuration will run two polling jobs:

- Job 1 will query **my1.sl1.com** using the defaults: poll every 60 seconds, starting from now
- Job 2 will query **my2.sl1.com** using overrides: poll every 120 seconds, starting from 09/05/2023, only query for Zebrium service groups **sg-1** and **sg-2**, send any unmatched events to SL1 device_id 1.

```
jobs:
                      # minimal config required job
                      # will default to all Zebrium Service Groups
                      # will drop all alerts that don't match an SL1 device
                      # polling will occur every 60s, starting from now
                      - name: example_job_1
                      ze_deployment_id: "sciencelogic_default"
                      sl1_api_config:
                      api_url: https://my1.sl1.com/
                      api_auth: "Basic dXNlcjpwYXNz"
                      # maximal config job
                      # will query only the 2 service groups provided
                      # will send any alerts that don't match an SL1 device to device
                      # will poll every 120 seconds from 9/5/2023 00:00:00 GMT to now
                      - name: example_job_2
                      sl1_default_device_ids:
                      - "1"
                      ze_service_groups:
                      - "sg-1"
                      - "sg-2"
                      ze_deployment_id: "some_other_deployment"
                      sl1_api_config:
                      api_url: https://my2.sl1.com/
                      api_auth: "Basic dXNlcjpwYXNz"
                      poll_timing:
                      poll_sleep_seconds: 120
                      poll_start_time_iso: "2023-09-05 00:00:00"
                sl1_override_event_time: false
```