



Monitoring Cisco Meraki (API)

Cisco: Meraki [API] PowerPack version 114

Table of Contents

Introduction	4
What is Cisco Meraki?	4
What Does the Cisco: Meraki [API] PowerPack Monitor?	5
Installing the Cisco: Meraki [API] PowerPack	5
Configuration and Discovery	7
Best Practices for Monitoring Meraki with SL1	8
Cisco: Meraki [API] Dynamic Applications Enabled and Alignment Status	9
Generating a Cisco Meraki API Key	11
Creating a Basic/Snippet Credential	14
Creating a Basic/Snippet Credential in the SL1 Classic User Interface	15
Creating a SOAP/XML Credential	16
Creating a SOAP/XML Credential in the SL1 Classic User Interface	20
Testing the Cisco Meraki API Credential	23
Testing the Cisco Meraki API Credential in the SL1 Classic User Interface	25
Creating a Cisco Meraki Virtual Device	26
Manually Aligning the Cisco: Meraki Organization Discovery Dynamic Application	27
Assigning a Device Class to a Discovered Device	27
Manually Aligning the Cisco: Meraki Organization Discovery Dynamic Application in the SL1 Classic User Interface	28
Bulk Unaligning a Dynamic Application from Devices	29
Configuring Dynamic Applications to Hide Empty Rows	29
Disabling the Encoding Fix in the Cisco: Meraki Request Manager [API] Dynamic Application	30
Updating the Polling Interval for the Cisco: Meraki Uplink Usage Performance [API] Dynamic Application	30
Viewing Cisco Meraki Component Devices	31
Creating Events from Cisco Meraki Emails	33
Formatting Inbound Emails	34
Enabling Inbound Email Alerts	35
Configuring Cisco: Meraki Webhooks	35
Managing the Cisco: Meraki Update Switch Configuration [API] Run Book Action Policy	39
Executing the Cisco: Meraki Update Switch Configuration [API] Run Book Action Policy Automatically	40
Enabling and Configuring the Alert	40

Manually Executing the Cisco: Meraki Update Switch Configuration [API] Run Book Action Policy	40
Managing the Cisco: Meraki Reboot Device Run Book Action Policy	41
Manually Executing the Cisco: Meraki Reboot Device Run Book Action Policy	42
Executing the Cisco: Meraki Reboot Device [API] Run Book Action Policy Automatically	42
Using Custom Device Classes with Cisco Meraki API	43
Creating a Custom Component Device Class	43
Adding Custom Device Classes to the PowerPack	44
Troubleshooting	44
Incorrect Calculations for Presentation Objects in the Cisco: Meraki Uplink Usage Performance [API] Dynamic Application	44
Cisco: Meraki Uplink Performance [API] retrieves a NULL value	45
Meraki Organizations are not modeling	45
Cisco: Meraki [API] API Endpoints	46
Cisco: Meraki [API] API Endpoints	46
Cisco: Meraki [API] Dynamic Application API Rates	47

Chapter

1

Introduction

Overview

This manual describes how to monitor Cisco Meraki access points, switches, firewalls, cameras, sensors, and other IOT devices in SL1 using the *Cisco: Meraki [API] PowerPack* and the Meraki API.

The following sections provide an overview of Cisco Meraki and the *Cisco: Meraki [API] PowerPack*:

This chapter covers the following topics:

<i>What is Cisco Meraki?</i>	4
<i>What Does the Cisco: Meraki [API] PowerPack Monitor?</i>	5
<i>Installing the Cisco: Meraki [API] PowerPack</i>	5

NOTE: ScienceLogic provides this documentation for the convenience of ScienceLogic customers. Some of the configuration information contained herein pertains to third-party vendor software that is subject to change without notice to ScienceLogic. ScienceLogic makes every attempt to maintain accurate technical information and cannot be held responsible for defects or changes in third-party vendor software. There is no written or implied guarantee that information contained herein will work for all third-party variants. See the End User License Agreement (EULA) for more information.

What is Cisco Meraki?

Cisco Meraki provides a set of networking devices and appliances that you can manage from the cloud. Cisco Meraki's centralized cloud architecture enables you to securely monitor users, applications, and devices in your environment.

What Does the Cisco: Meraki [API] PowerPack Monitor?

To monitor Cisco Meraki devices using SL1 and the Meraki API, you must install the *Cisco: Meraki [API]* PowerPack. This PowerPack enables you to discover and collect data about Cisco Meraki appliances.

The *Cisco: Meraki [API]* PowerPack includes:

- Dynamic Applications to discover, model, and monitor performance metrics and collect configuration data for Cisco Meraki devices
- Device Classes for each of the Cisco Meraki devices, networks, and organizations that SL1 monitors
- Event Policies and corresponding alerts that are triggered when Cisco Meraki devices meet certain status criteria
- Event Policies for Email/Webhook alerts
- Example credentials that you can use as template to create Basic/Snippet or SOAP/XML credentials for connecting to the Cisco Meraki API
- Run Book Action and Automation policies that create Meraki organization virtual devices during discovery, reboot devices, and vanish devices

NOTE: Meraki dashboards can be downloaded in the Cisco Meraki: SL1 Dashboards PowerPack.

Installing the Cisco: Meraki [API] PowerPack

Before completing the steps in this manual, you must import and install the latest version of the *Cisco: Meraki [API]* PowerPack.

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 section on [Global Settings](#).

To download and install the PowerPack:

1. Search for and download the PowerPack from the **PowerPacks** page (Product Downloads > PowerPacks & SyncPacks) at the [ScienceLogic Support Site](#).
2. In SL1, go to the **PowerPacks** page (System > Manage > PowerPacks).
3. Click the **[Actions]** button and choose *Import PowerPack*. The **Import PowerPack** dialog box appears.
4. Click **[Browse]** and navigate to the PowerPack file from step 1.
5. Select the PowerPack file and click **[Import]**. The **PowerPack Installer** modal displays a list of the PowerPack contents.
6. Click **[Install]**. The PowerPack is added to the **PowerPacks** page.

NOTE: If you exit the **PowerPack Installer** modal without installing the imported PowerPack, the imported PowerPack will not appear in the **PowerPacks** 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

2

Configuration and Discovery

Overview

The following sections describe how to configure and discover Cisco Meraki devices for monitoring by SL1 using the *Cisco: Meraki [API]* PowerPack and the Meraki API:

This chapter covers the following topics:

<i>Best Practices for Monitoring Meraki with SL1</i>	8
<i>Generating a Cisco Meraki API Key</i>	11
<i>Creating a Basic/Snippet Credential</i>	14
<i>Creating a SOAP/XML Credential</i>	16
<i>Testing the Cisco Meraki API Credential</i>	23
<i>Creating a Cisco Meraki Virtual Device</i>	26
<i>Manually Aligning the Cisco: Meraki Organization Discovery Dynamic Application</i>	27
<i>Viewing Cisco Meraki Component Devices</i>	31
<i>Creating Events from Cisco Meraki Emails</i>	33
<i>Configuring Cisco: Meraki Webhooks</i>	35
<i>Managing the Cisco: Meraki Update Switch Configuration [API] Run Book Action Policy</i>	39
<i>Managing the Cisco: Meraki Reboot Device Run Book Action Policy</i>	41
<i>Using Custom Device Classes with Cisco Meraki API</i>	43
<i>Troubleshooting</i>	44

Best Practices for Monitoring Meraki with SL1

Cisco Meraki differs from other solutions on the market. Meraki is focused on abstraction of complexities and ease of use. As a result, Meraki devices do not expose anywhere near the same level of data as typical network technologies. Because of this, expectations of data reported and methods of monitoring must differ from other technologies that cater to “Power Users”.

The Cisco Meraki API integration is expected to be used in conjunction with email or webhook ingestion of alerts from the Meraki dashboard into SL1. ScienceLogic's API integration collection purposely does not replicate data collection for metrics that exist as an alarm in the Meraki dashboard to avoid “wasting” the set amount of API calls SL1 can make to the Meraki dashboard API within the API Rate limit. See the sections below on configuring inbound alerts from Meraki via email or webhooks. See the Alerts and Notifications section of the [Cisco Meraki documentation](#) to view all of the alerts possible in the Meraki dashboard.

NOTE: The Cisco Meraki API rate limit is determined by how many API calls are made for a Cisco Meraki organization, regardless of the user, API key, or IP address making the API call. As a result, ScienceLogic does not recommend using multiple SL1 systems to monitor a single Cisco Meraki organization, or using any other tools that utilize the Cisco Meraki API for the same organization at the same time.

Meraki does not expose much, if any of the typical “performance” data you might expect in any of the bulk API calls offered. This data is not intended to be used for monitoring and is designed to only be fetched on a per-device, as needed basis. The integration does include some of this data collection, but the Dynamic Applications that do so are turned off by default and/or not aligned to devices in order to avoid taxing the API rate limit for large customers with other tools also using the Meraki API.

Additionally, Meraki does not expose CPU, Memory, Swap, or other typical “vital” metrics as you would expect with other networking gear and therefore, SL1 does not collect it. If you want, you can attempt to monitor Meraki devices via SNMP with an IP based discovery, but the management information bases on Meraki devices are very limited and not much data will be collected. ScienceLogic does not recommend discovering Meraki devices via IP address and merging them with the API based components in this PowerPack, since Meraki requires DHCP to be enabled, and if IP addresses are reassigned after merging the devices, collection issues will occur.

Some Dynamic Applications may align to devices that you do not expect data to be collected for as a method of future proofing the integration in case the API is updated to respond with data for those devices. This is only done when the API call required to collect the data can be done for all devices in a Meraki Organization at once, and it's “free” to align it to all devices.

The devices listed below are generally expected to have data for the following Dynamic Applications if the device it is aligned to has data to return, is enabled, and has the proper Dynamic Application and credential aligned. In general, only Dynamic Applications that use a bulk API call across an entire Meraki Organization are enabled and aligned by default. Dynamic Applications not enabled and aligned by default are generally using API calls that are per network or per device and will utilize many more API calls to collect data than other collections. Meraki only allows 10 API calls per Organization, per requesting IP address per second.

Cisco: Meraki [API] Dynamic Applications Enabled and Alignment Status

Certain Dynamic Applications in the Cisco: Meraki [API] PowerPack are disabled by default. In order for these Dynamic Applications to begin collecting data, they must first be enabled. Others need to be manually aligned in order to collect data.

The table below displays the list of Dynamic Applications and whether they need to be manually enabled or aligned. For more information about the API endpoints used by each Dynamic Application and their API Rates, see the Cisco: Meraki [API] API Endpoints Appendix.

Dynamic Applications	Enabled by Default?	Aligned by Default?	Aligning Enabling / Unaligning Disabling saves API calls?	Devices Expected to Collect Data	Required for Basic Device Discovery?
Cisco: Meraki Organization Discovery [API]	Yes	Yes, Meraki Cloud Controller	Yes	Meraki Cloud Controller	Required to discover new Organizations or update existing ones
Cisco: Meraki API HTTP Stats [API]	Yes	Yes, Meraki organization	No	Meraki organization	No
Cisco: Meraki Component Counts [API]	Yes	Yes, Meraki organization	No	Meraki organization	No
Cisco: Meraki Network Discovery [API]	Yes	Yes, Meraki organization	No	Meraki organization	Required to discover new Networks and to populate the cache for the "Cisco: Meraki Network Configuration [API]" Dynamic Application
Cisco: Meraki Organization License Configuration [API]	Yes	Yes, Meraki organization	No	Meraki organization	No
Cisco: Meraki Request Manager [API]	Yes	Yes, Meraki organization	Yes, but Do Not Disable	Meraki organization	Required. Every API call is made by the Request Manager when it runs
Cisco: Meraki Device Discovery [API]	Yes	Yes, Meraki network	No	Meraki network	Required
Cisco: Meraki Network Component Counts [API]	Yes	Yes, Meraki network	No	Meraki network	No

Dynamic Applications	Enabled by Default?	Aligned by Default?	Aligning Enabling / Unaligning Disabling saves API calls?	Devices Expected to Collect Data	Required for Basic Device Discovery?
Cisco: Meraki Network Configuration [API]	Yes	Yes, Meraki network	No	Meraki network	No
Cisco: Meraki Device Configuration [API]	Yes	Yes, all Meraki devices	No	Meraki devices	No
Cisco: Meraki Uplink Usage Performance [API]	No	No	Yes, but if aligned to a device, the API call will be made for every device in the organization.	Devices where Meraki returns data for uplinks (Usually MX and Z)	No
Cisco: Meraki Uplink Performance [API]	Yes	Yes, all Meraki devices	No	Devices where Meraki returns data for uplinks (Usually MX and Z)	No
Cisco: Meraki Uplink Status [API]	Yes	Yes, all Meraki devices	No	Devices where Meraki returns data for uplinks (Usually MX and Z)	No
Cisco: Meraki VPN Status [API]	Yes	No	Yes, but if aligned to a device, the API call will be made for every device in the organization.	Devices where Meraki returns data for VPN connectivity (Usually MX and Z)	No
Cisco: Meraki Switch Ports Configuration [API]	Yes	Yes, all Meraki devices	Yes, but if aligned to a device, the API call will be made for every device in the organization.	Devices where Meraki returns data for Switches	No
Cisco: Meraki Switch Ports Status Configuration [API]	No	No	Yes, but if aligned to a device, the API call will be made for every device in the organization.	Devices where Meraki returns data for Switches	No

Dynamic Applications	Enabled by Default?	Aligned by Default?	Aligning Enabling / Unaligning Disabling saves API calls?	Devices Expected to Collect Data	Required for Basic Device Discovery?
Cisco: Meraki Switch Port Status Performance [API]	No	No	No	Devices where the "Switch Ports Status Configuration [API]" Dynamic Application is aligned and collecting data	No

NOTE: If you want to disable or unalign the "Cisco: Meraki Switch Ports Status Configuration [API]" Dynamic Application, the "Cisco: Meraki Switch Ports Status Performance [API]" Dynamic Application should be disabled as well. If the "Cisco: Meraki Switch Ports Status Configuration [API]" Dynamic Application is disabled, the "Cisco: Meraki Switch Ports Status Performance [API]" Dynamic Application will not have data to read from the cache and will not display any data.

Generating a Cisco Meraki API Key

To configure Cisco Meraki for monitoring using the Meraki API, you must first generate an API key for a read-only Meraki user. You will then enter this user's API key in the [Basic/Snippet credential](#) or [SOAP/XML credential](#) you create in SL1 to monitor Meraki.

IMPORTANT: While an API key for a read-only Meraki user is acceptable for most credential purposes in this PowerPack, an API key for a user with write permissions is required for the automation that changes configurations and reboots devices.

NOTE: If the read-only user has access to multiple organizations, then SL1 can discover all of those organizations with a single discovery session. In this scenario, each organization is created as a separate "root level" device with the networks and devices for that organization modeled in the DCM Tree as children of the organization. Organizations and their child components can be moved between collectors after discovery for load balancing purposes.

However, if you want each Meraki organization to have its own corresponding ScienceLogic organization in SL1, ScienceLogic recommends creating a unique read-only user account and API key for each organization in Meraki. You can then create separate credentials in SL1 for each Meraki organization using those unique API keys, and then use those credentials to run separate discovery sessions for each organization.

To create a read-only user:

1. Log in to the Cisco Meraki web interface.
2. Go to **Organization > Administrators**, and then click the **[Add admin]** button.
3. On the **Create administrator** page, complete the following fields:

The screenshot shows a modal window titled "Create administrator" with a close button (X) in the top right corner. The form contains the following fields and elements:

- Name:** A text input field.
- Email:** A text input field.
- Organization access:** A dropdown menu currently set to "Read-only".
- Access Privileges Table:** A table with two columns: "Target" and "Access". Below the table is a link that says "+ Add access privileges".
- Footer:** A link for "privacy" on the left, and two buttons, "Close" and "Create admin", on the right.

- **Name.** Type the user's name.
 - **Email.** Type the user's email address.
 - **Organization access.** Select *Read-only*.
4. Click **[Create admin]**. Cisco Meraki sends an email to the email address provided, describing how the user can complete the registration process. The user must complete those steps before generating the API key.

To generate a Cisco Meraki API key for that read-only user:

1. Log in to the Cisco Meraki web interface as the read-only user.

2. Go to **Organization > Settings:**

SNMP

Version 2C SNMP V2C disabled

Version 3 SNMP V3 disabled

IP restrictions

Enter IP addresses separated by whitespace, commas, or semicolons. Leave blank to allow SNMP queries from all IP addresses.

Dashboard API access

API Access

☒ Enable access to the Cisco Meraki Dashboard API

After enabling the API here, go to your [profile](#) to generate an API key. The API will return 404 for requests with a missing or incorrect API key.

Delete this organization

You can delete this organization only if it has no networks, users, licenses, or devices claimed in its inventory.

[Delete organization](#)

[Save Changes](#) [cancel](#)

(Please allow 1-2 minutes for changes to take effect.)

© 2017 Cisco Systems, Inc. [privacy](#) [terms](#)

Last login: 10 minutes ago from your current IP address
Current session started: 10 minutes ago

[Make a wish](#)

3. In the **Dashboard API access** section, select the **Enable access to the Cisco Meraki Dashboard API** checkbox.
4. Click the **Save Changes** button.
5. Click the **profile** link in the **Dashboard API access** section.
6. In your user profile, navigate to the **API access** section and click the **Generate new API key** button.

Your account

[View](#) your account settings.

Change your password

Current password ••••••••

New password

Confirm password

[Change password](#)

Two-factor authentication

SMS authentication is OFF.

[Set up SMS authentication](#)

You can also set up offline access on a mobile device: [Android](#), [iPhone](#), or [BlackBerry](#)

API access

API keys

[Generate new API key](#)

Color blind assist mode (OFF)

Enables an alternative color palette for various Dashboard elements.

[Enable Red/Green assist mode](#)

Sample of elements affected by color blind assist mode:

Device status icons: Active: Active Alerting: Alerting Unreachable: Unreachable Dormant: Dormant

Map pins: Gateway Repeater Alerting Offline

Connectivity: Connectivity

Connectivity icons: Connectivity

Labels: Success Alert

7. In the **API access** section, the API key appears. Copy and save the key value.



NOTE: API keys are visible only to the user that created them.

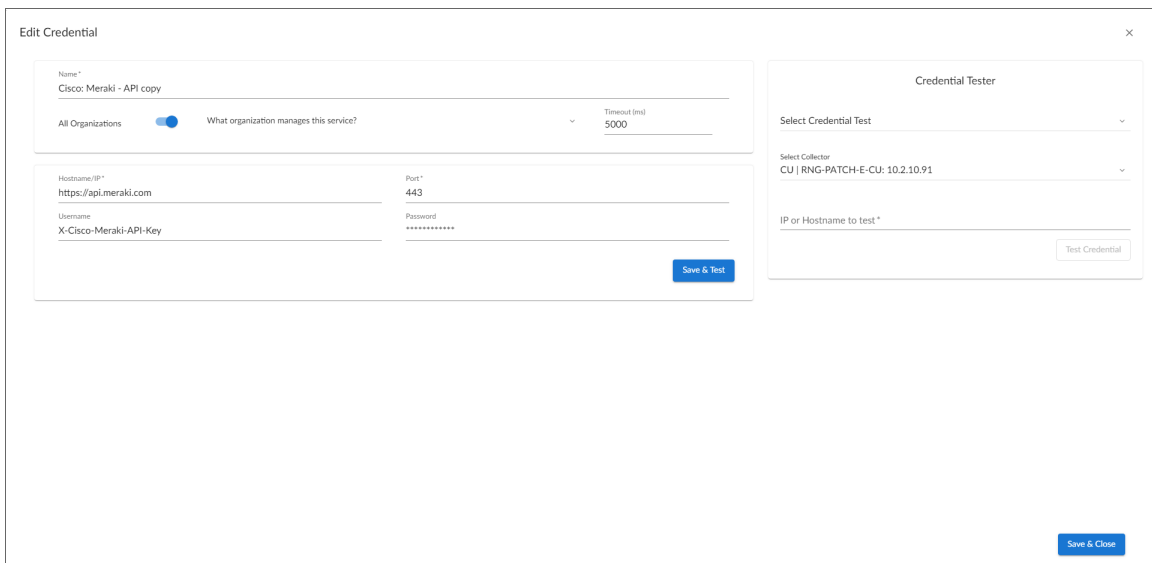
Creating a Basic/Snippet Credential

To configure SL1 to monitor Cisco Meraki systems using the Meraki API, you must create a Basic/Snippet credential. This credential allows the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack to connect with the Cisco Meraki API. An example Basic/Snippet credential that you can edit for your own use is included in the PowerPack.

NOTE: If you are using an SL1 system prior to version 11.1.0, the new user interface does not include the **Duplicate** option for sample credential(s). ScienceLogic recommends that you use [the classic user interface and the Save As button](#) to create new credentials from sample credentials. This will prevent you from overwriting the sample credential(s).

To create a Basic/Snippet credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the **Cisco: Meraki - API** sample credential, click its **[Actions]** icon () and select **Duplicate**. A copy of the credential, called **Cisco: Meraki - API copy** appears.
3. Click the **[Actions]** icon () for the **Cisco: Meraki - API copy** credential and select **Edit**. The **Edit Credential** page appears:



4. Supply values in the following fields:
 - **Name.** Type a new name for the Meraki 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.
- **Timeout (ms).** Keep the default value.
- **Hostname/IP.** Keep the default value.

NOTE: You **must** use the default value of "https://api.meraki.com" in the **Hostname/IP** field.

- **Port.** Keep the default value.
- **Username.** Keep the default value.
- **Password.** Type the [Meraki API key](#).

4. Click **[Save & Close]**.

NOTE: If you would like to test your credential using the Credential Tester panel, click **[Save & Test]**. For detailed instructions on using the Credential Tester panel, see the [Testing the Cisco Meraki API Credential](#) section.

Creating a Basic/Snippet Credential in the SL1 Classic User Interface

To configure SL1 to monitor Cisco Meraki systems using the Meraki API, you must create a Basic/Snippet credential. This credential allows the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack to connect with the Cisco Meraki API. An example Basic/Snippet credential that you can edit for your own use is included in the PowerPack.

To create a Basic/Snippet credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the **Cisco: Meraki - API** credential, and then click its wrench icon (🔧). The **Edit Basic/Snippet Credential** modal page appears:

The screenshot shows a web-based form titled "Credential Editor [130]". Inside, there's a sub-header "Edit Basic/Snippet Credential #130" with "New" and "Reset" buttons. The form is divided into a "Basic Settings" section. It contains several input fields: "Credential Name" with the value "Cisco: Meraki - API", "Hostname/IP" with "https://api.meraki.com", "Port" with "443", "Timeout(ms)" with "5000", "Username" with "X-Cisco-Meraki-API-Key", and a "Password" field filled with dots. At the bottom of the form are "Save" and "Save As" buttons.

3. Complete the following fields:

- **Credential Name.** Type a new name for the credential.
- **Hostname/IP.** Keep the default value.

NOTE: You **must** use the default value of "https://api.meraki.com" in the **Hostname/IP** field.

- **Port.** Keep the default value.
- **Timeout(ms).** Keep the default value.
- **Username.** Keep the default value.
- **Password.** Type the [Meraki API key](#).

4. Click the **[Save As]** button.
5. When the confirmation message appears, click **[OK]**.

Creating a SOAP/XML Credential

If you access Meraki systems through a third-party proxy server, you can create a SOAP/XML credential to enable the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack to connect with the Cisco Meraki API via the proxy server.

Similarly, if you want to discover only some selected devices, you can create a SOAP/XML credential that specifies tag values that the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack can use to determine which devices should be discovered.

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).

Two example SOAP/XML credentials that you can edit for your own use are included in the PowerPack:

- **Cisco: Meraki - API - Proxy**, for users who connect to Meraki through a third-party proxy server

Name *

Cisco: Meraki - API - Proxy

All Organizations

What organization manages this service?

Timeout (ms)

5000

Content Encoding

text/xml

Method

POST

HTTP Version

http/1.1

URL

https://api.meraki.com

HTTP Auth User

X-Cisco-Meraki-API-Key

HTTP Auth Password

Hostname/IP

10.0.0.0

Port (number optional)

3128

User

em7admin

Password

Embedded Password (N/A)

Embed Value [%1]

Embed Value [%2]

Embed Value [%3]

Embed Value [%4]

HTTP Headers

Add Header

Save & Close

Credential Tester

Select Credential Test

Select Collector

CU | RNG-PATCH-E-CU: 10.2.10.91

IP or Hostname to test *

Test Credential

- **Cisco: Meraki - API (Selective)**, for users who want to discover only some selected devices based on tag values

Name *

Cisco: Meraki - API (Selective)

All Organizations

What organization manages this service?

Timeout (ms)

5000

Content Encoding

text/xml

Method

POST

HTTP Version

http/1.1

URL

https://api.meraki.com

HTTP Auth User

X-Cisco-Meraki-API-Key

HTTP Auth Password

Hostname/IP

0

Port (number optional)

User

Password

Embedded Password (N/A)

Embed Value [%1]

Embed Value [%2]

Embed Value [%3]

Embed Value [%4]

HTTP Headers

Add Header

Save & Close

Credential Tester

Select Credential Test



Select Collector

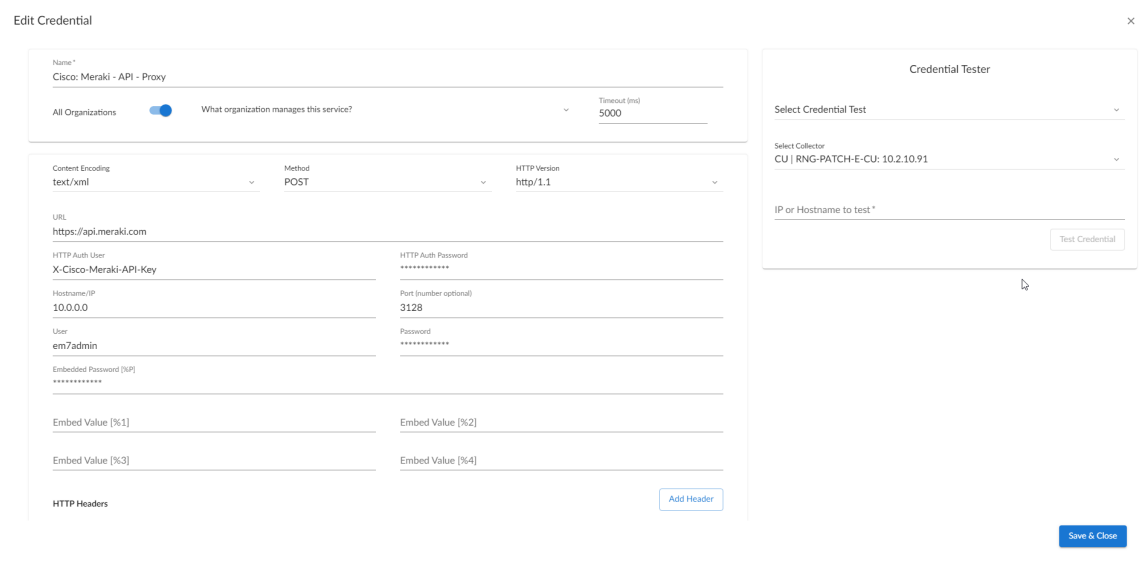
CU | RNG-PATCH-E-CU: 10.2.10.91

IP or Hostname to test *

Test Credential

To define a SOAP/XML credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the sample credential you want to use, then click its **[Actions]** icon () and select **Duplicate**. A copy of the credential, called **Cisco: Meraki - API - Proxy copy** or **Cisco: Meraki - API (Selective) copy** appears.
3. Click the **[Actions]** icon () for the credential copy and select **Edit**. The **Edit Credential** modal page appears.



3. Supply values in the following fields:
 - **Name**. Type a new name for your Meraki 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.
 - **Timeout (ms)**. Keep the default value.
 - **Content Encoding**. Keep the default value.
 - **Method**. Keep the default value.
 - **HTTP Version**. Keep the default value.
 - **URL**. Keep the default value of "https://api.meraki.com".
 - **HTTP Auth User**. Keep the default value.
 - **HTTP Auth Password**. Type the [Meraki API key](#).

Proxy Settings

NOTE: You must complete the **Proxy Settings** fields only if you connect to the Meraki API through a third-party proxy server. If you do not use a proxy to connect to Meraki, then you can leave these fields blank.

- **Hostname/IP.** Type the server's hostname or IP address.
- **Port.** Type the port on the proxy server to which you will connect.
- **User.** Type the username used to access the proxy server.
- **Password.** Type the password used to access the proxy server.

HTTP Headers

- **proxy_url_protocol:http.** Edit this header if you want to connect a proxy using a different protocol, such as http or https. The default value is "http".
- **Add a header.** Click **[Add a header]** once if you want to include tag values for SL1 to match when it discovers Meraki devices, or click **[Add a header]** twice if you want to include tag values and specify that tag-matching should be case-insensitive. In the blank fields that appear, do one or both of the following:
 - Type "tags:" in the first field, followed by one or more tag values. You can include multiple tag values in a string, using comma separators and no spaces. For example: "tags:value1,value2,value3".
 - Type "regex:IGNORECASE" in the second field if you want SL1 to match the tag values regardless of case.

NOTE: If you are using a tag to discover a device and want to discover that device's network, the device and its network must have the same tag applied.

NOTE: Tag values can include wildcard characters.

NOTE: After initial discovery, you can add more tag values and run discovery again to discover additional component devices. However, if you remove tag values and then run discovery again, the component devices that had been discovered based on the removed tag values will be updated to an unavailable state.

4. Click **[Save & Close]**.

NOTE: If you would like to test your credential using the Credential Tester panel, click **[Save & Test]**. For detailed instructions on using the Credential Tester panel, see the [Testing the Cisco Meraki API Credential](#) section.

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

If you access Meraki systems through a third-party proxy server, you can create a SOAP/XML credential to enable the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack to connect with the Cisco Meraki API via the proxy server.

Similarly, if you want to discover only some selected devices, you can create a SOAP/XML credential that specifies tag values that the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack can use to determine which devices should be discovered.

Two example SOAP/XML credentials that you can edit for your own use are included in the PowerPack:

- **Cisco: Meraki - API - Proxy**, for users who connect to Meraki through a third-party proxy server

The screenshot shows the 'Credential Editor [87]' window with the following sections:

- Edit SOAP/XML Credential #87** (Title bar)
- Basic Settings**
 - Profile Name: Cisco: Meraki - API - Proxy
 - Content Encoding: [text/xml]
 - Method: [POST]
 - HTTP Version: [HTTP/1.1]
 - URL [http(s)://Host:Port/Path | %D = Aligned Device Address | %N = Aligned Device Host Name]: https://api.meraki.com
 - HTTP Auth User: X-Cisco-Meraki-API-Key
 - HTTP Auth Password: ...
 - Timeout (seconds): 5
- Proxy Settings**
 - Hostname/IP: 10.0.0.0
 - Port: 3128
 - User: em7admin
- CURL Options**
 - CAINFO
 - CAPATH
 - CLOSEPOLICY
 - CONNECTTIMEOUT
 - COOKIE
 - COOKIEFILE
 - COOKIEJAR
 - COOKIELIST
 - CRLF
 - CUSTOMREQUEST
 - DNSCACHETIMEOUT
- Soap Options**
 - Embedded Password [%P]
 - Embed Value [%1]
 - Embed Value [%2]
 - Embed Value [%3]
 - Embed Value [%4]
- HTTP Headers**
 - + Add a header
 - proxy_url_protocol:http
- Buttons**: Save, Save As

- **Cisco: Meraki - API (Selective)**, for users who want to discover only some selected devices based on tag values

Credential Editor [88]

Edit SOAP/XML Credential #88 [New] [Reset]

Basic Settings

Profile Name: Cisco: Meraki - API (Selective) Content Encoding: [text/xml] Method: [POST] HTTP Version: [HTTP/1.1]

URL [http(s)://Host:Port/Path | %D = Aligned Device Address | %N = Aligned Device Host Name]
https://api.meraki.com

HTTP Auth User: X-Cisco-Meraki-API-Key HTTP Auth Password: ***** Timeout (seconds): 5

Proxy Settings

Hostname/IP: Port: 0 User:

CURL Options

CAINFO CAPATH CLOSEPOLICY CONNECTTIMEOUT COOKIE COOKIEFILE COOKIEJAR COOKIELIST CRLF CUSTOMREQUEST DNSCACHETIMEOUT

Soap Options

Embedded Password [%P]

Embed Value [%1] Embed Value [%2]

Embed Value [%3] Embed Value [%4]

HTTP Headers

+ Add a header

tags:Sciencel?gic

regex:IGNORECASE

[Save] [Save As]

To define a SOAP/XML credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the Cisco Meraki example credential that you want to use and click its wrench icon (🔧). The **Credential Editor** modal page appears:

Credential Editor [88]

Edit SOAP/XML Credential #88 [New] [Reset]

Basic Settings

Profile Name: Cisco: Meraki - API (Selective) Content Encoding: [text/xml] Method: [POST] HTTP Version: [HTTP/1.1]

URL [http(s)://Host:Port/Path | %D = Aligned Device Address | %N = Aligned Device Host Name]
https://api.meraki.com

HTTP Auth User: X-Cisco-Meraki-API-Key HTTP Auth Password: ***** Timeout (seconds): 5

Proxy Settings

Hostname/IP: Port: 0 User:

CURL Options

CAINFO CAPATH CLOSEPOLICY CONNECTTIMEOUT COOKIE COOKIEFILE COOKIEJAR COOKIELIST CRLF CUSTOMREQUEST DNSCACHETIMEOUT

Soap Options

Embedded Password [%P]

Embed Value [%1] Embed Value [%2]

Embed Value [%3] Embed Value [%4]

HTTP Headers

+ Add a header

tags:Sciencel?gic

regex:IGNORECASE

[Save] [Save As]

3. Enter values in the following fields:

Basic Settings

- **Profile Name.** Type a new name for your Meraki credential.
- **HTTP Auth Password.** Type the [Meraki API key](#).

NOTE: You can use the default values for the remaining **Basic Settings** fields. You **must** use the default value in the **URL** field.

Proxy Settings

NOTE: You must complete the **Proxy Settings** fields only if you connect to the Meraki API through a third-party proxy server. If you do not use a proxy to connect to Meraki, then you can leave these fields blank.

- **Hostname/IP.** Type the server's hostname or IP address.
- **Port.** Type the port on the proxy server to which you will connect.
- **User.** Type the username used to access the proxy server.
- **Password.** Type the password used to access the proxy server.

HTTP Headers

- **proxy_url_protocol:http.** Edit this header if you want to connect a proxy using a different protocol, such as http or https. The default value is "http".
- **Add a header.** Click **[Add a header]** once if you want to include tag values for SL1 to match when it discovers Meraki devices, or click **[Add a header]** twice if you want to include tag values and specify that tag-matching should be case-insensitive. In the blank fields that appear, do one or both of the following:
 - Type "tags:" in the first field, followed by one or more tag values. You can include multiple tag values in a string, using comma separators and no spaces. For example: "tags:value1,value2,value3".
 - Type "regex:IGNORECASE" in the second field if you want SL1 to match the tag values regardless of case.

NOTE: If you are using a tag to discover a device and want to discover that device's network, the device and its network must have the same tag applied.

NOTE: Tag values can include wildcard characters.

NOTE: After initial discovery, you can add more tag values and run discovery again to discover additional component devices. However, if you remove tag values and then run discovery again, the component devices that had been discovered based on the removed tag values will be updated to an unavailable state.

4. Click the **[Save As]** button, and then click **[OK]**.

Testing the Cisco Meraki API Credential

The *Cisco: Meraki [API]* PowerPack includes a credential test for Cisco Meraki credentials. Credential tests define a series of steps that SL1 can execute on demand to validate whether a credential works as expected.

The Cisco Meraki credential tests can be used to test the Basic/Snippet and SOAP/XML credentials for monitoring the Cisco Meraki API using the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack.

The **Cisco: Meraki [API] (Basic/Snippet) Credential tester** performs the following steps:

- **Test Meraki Reachability.** Performs an ICMP ping request to the host specified in the credential.
- **Test Meraki Port Availability.** Performs an NMAP request to the TCP port specified in the credential on the host specified in the credential.
- **Test Meraki Organization Request.** Performs a check to see if the Meraki organization request has been collected appropriately.

The **Cisco: Meraki [API] (SOAP/XML) Credential tester** performs the following steps:

- **Test Meraki Reachability.** Performs an ICMP ping request to the host specified in the credential.
- **Test Meraki Port Availability.** Performs an NMAP request to the TCP port specified in the credential on the host specified in the credential.
- **Test Meraki Organization Request.** Performs a check to see if the Meraki organization request has been collected appropriately.

To test the Cisco Meraki credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the credential you wish to test, select the **Actions** button (⋮) next to it and click *Edit/Test*. The **Edit Credential** modal page appears:

Name *

Cisco: Meraki - API

All Organizations

Select the organizations the credential belongs to *

Timeout (ms)

5000

Hostname/IP *

https://api.meraki.com

Port *

443

Username

X-Cisco-Meraki-API-Key

Password

Credential Tester

Select Credential Test

Cisco: Meraki [API] (Basic/Snippet) Credential tester

Select Collector

CUG1 | RING-MIG-B-CU1: 10.2.10.32

IP or Hostname to test *

api.meraki.com

Test Credential

Close

- In the **Credential Tester** pane on the right, fill out the following fields on this page:
 - **Select Credential Test.** Select **Cisco: Meraki [API] (Basic/Snippet) Credential tester** or the **Cisco: Meraki [API] (SOAP/XML) Credential tester**, depending on which credential you are testing.
 - **Collector.** Select the All-In-One Appliance or Data Collector that will run the test.
 - **IP or Hostname to Test.** Enter "api.meraki.com" or the IP address of your Meraki system.
- Click the **[Run Test]** button to run the credential test. The **Testing Credential** window appears.

Testing Credential

STEP	DESCRIPTION	LOG MESSAGE	STATUS
Test Meraki reachability.	Check to see if the IP/Hostname is reachable using ICMP.	The state of IP/Hostname [api.meraki.com] is 'reachable' using ICMP, t...	✓ Passed ?
Test Meraki port availa...	Check to see if the Meraki port has been open appropriately.	The IP/Hostname [api.meraki.com] is using the port [443], the current...	✓ Passed ?
Test Meraki organizati...	Check to see if the Meraki organizations request has been collected a...	Collected: 4 Organizations.	✓ Passed ?

The **Testing Credential** window displays a log entry for each step in the credential test. The steps performed are different for each credential test. The log entry for each step includes the following information:

- **Step.** The name of the step.
- **Description.** A description of the action performed during the step.
- **Log Message.** The result of the step for this execution of the credential test.

- **Status.** Whether the result of this step indicates the credential and/or the network environment is configured correctly (Passed) or incorrectly (Failed).
- **Step Tip.** Mouse over the question mark icon (?) to display the tip text. The tip text recommends what to do to change the credential and/or the network environment if the step has a status of "Failed".

Testing the Cisco Meraki API Credential in the SL1 Classic User Interface

The *Cisco: Meraki [API]* PowerPack includes a credential test for Cisco Meraki credentials. Credential tests define a series of steps that SL1 can execute on demand to validate whether a credential works as expected.

The Cisco Meraki credential tests can be used to test the Basic/Snippet and SOAP/XML credentials for monitoring the Cisco Meraki API using the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack.

The **Cisco: Meraki [API] (Basic/Snippet) Credential tester** performs the following steps:

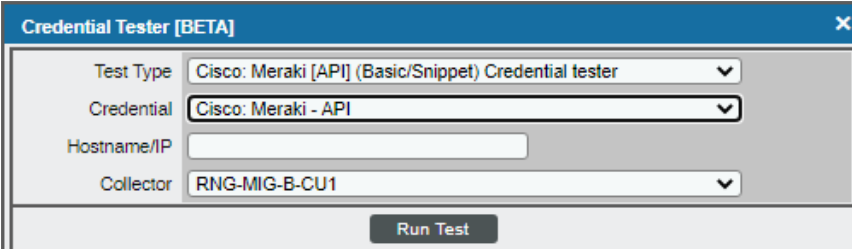
- **Test Meraki Reachability.** Performs an ICMP ping request to the host specified in the credential.
- **Test Meraki Port Availability.** Performs an NMAP request to the TCP port specified in the credential on the host specified in the credential.
- **Test Meraki Organization Request.** Performs a check to see if the Meraki organization request has been collected appropriately.

The **Cisco: Meraki [API] (SOAP/XML) Credential tester** performs the following steps:

- **Test Meraki Reachability.** Performs an ICMP ping request to the host specified in the credential.
- **Test Meraki Port Availability.** Performs an NMAP request to the TCP port specified in the credential on the host specified in the credential.
- **Test Meraki Organization Request.** Performs a check to see if the Meraki organization request has been collected appropriately.

To test the Cisco Meraki credential:

1. Go to the **Credential Test Management** page (System > Customize > Credential Tests).
2. Locate the **Cisco: Meraki [API] (Basic/Snippet) Credential tester** or the **Cisco: Meraki [API] (SOAP/XML) Credential tester**, depending on which credential you are testing, and click its lightning bolt icon (⚡). The **Credential Tester** modal page appears:

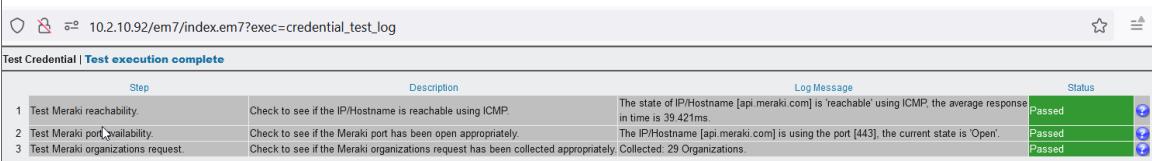


The screenshot shows a modal window titled "Credential Tester [BETA]" with a close button (X) in the top right corner. Inside the modal, there are four labeled fields: "Test Type" with a dropdown menu showing "Cisco: Meraki [API] (Basic/Snippet) Credential tester", "Credential" with a dropdown menu showing "Cisco: Meraki - API", "Hostname/IP" with an empty text input field, and "Collector" with a dropdown menu showing "RNG-MIG-B-CU1". At the bottom center of the modal is a button labeled "Run Test".

3. Supply values in the following fields:

- **Test Type.** This field is pre-populated with the credential test you selected.
- **Credential.** Select the credential to test. This drop-down list includes only credentials that you have access to that can be tested using the selected credential test.
- **Hostname/IP.** Enter "api.meraki.com" or the IP address of your Meraki system.
- **Collector.** Select the All-In-One Appliance or Data Collector that will run the test.

4. Click the **[Run Test]** button to run the credential test. The **Test Credential** window appears.



The screenshot shows a web browser window with the URL 10.2.10.92/em7/index.em7?exec=credential_test_log. The page title is "Test Credential | Test execution complete". Below the title is a table with four columns: Step, Description, Log Message, and Status. The table contains three rows of test results, all of which are "Passed".

Step	Description	Log Message	Status
1 Test Meraki reachability.	Check to see if the IP/Hostname is reachable using ICMP.	The state of IP/Hostname [api.meraki.com] is 'reachable' using ICMP, the average response in time is 39.421ms.	Passed
2 Test Meraki port availability.	Check to see if the Meraki port has been open appropriately.	The IP/Hostname [api.meraki.com] is using the port [443], the current state is 'Open'.	Passed
3 Test Meraki organizations request.	Check to see if the Meraki organizations request has been collected appropriately.	Collected: 29 Organizations.	Passed

The **Test Credential** window displays a log entry for each step in the credential test. The steps performed are different for each credential test. The log entry for each step includes the following information:

- **Step.** The name of the step.
- **Description.** A description of the action performed during the step.
- **Log Message.** The result of the step for this credential test.
- **Status.** Whether the result of this step indicates the credential or the network environment is configured correctly (Passed) or incorrectly (Failed).
- **Step Tip.** Mouse over the question mark icon (?) to display the tip text. The tip text recommends what to do to change the credential or the network environment if the step has a status of "Failed".

Creating a Cisco Meraki Virtual Device

To monitor your Cisco Meraki devices, you must create a **virtual device** that represents the Meraki Cloud Controller. A virtual device is a user-defined container that represents a device or service that cannot be discovered by SL1. You can use the virtual device to store information gathered by policies or Dynamic Applications.

If you want to discover more than one Meraki account, you must create a virtual device for each API key that you want to use.

To create a virtual device that represents your Meraki Cloud Controller:

1. Go to the **Device Manager** page (Devices > Device Manager).
2. Click **[Actions]** and select *Create Virtual Device* from the menu. The **Virtual Device** modal page appears.
3. Enter values in the following fields:
 - **Device Name.** Enter a name for the device.

- **Organization.** Select the organization for this device. The organization you associate with the device limits the users that will be able to view and edit the device. Typically, only members of the organization will be able to view and edit the device.
 - **Device Class.** Select *Cisco Systems | Meraki Cloud Controller*.
 - **Collector.** Select the collector group that will monitor the device.
4. Click **[Add]** to create the virtual device.
 5. Repeat these steps for each Meraki API key that you want to use.

Manually Aligning the Cisco: Meraki Organization Discovery Dynamic Application

After creating the Cisco Meraki virtual device, you must manually align the "Cisco: Meraki Organization Discovery [API]" Dynamic Application to the Cisco Meraki virtual device.

To manually align the Cisco Meraki Dynamic Application:

1. Go to the **Devices** page (Devices > Device Manager).
2. Locate your Cisco Meraki virtual device and click its name.
3. In the **Device Investigator**, click the **[Collections]** tab.
4. Click the **[Edit]** button at the top of the page, then click the **[Align Dynamic App]** button.
5. In the **Align Dynamic Application** modal, click **Choose Dynamic Application**.
6. Locate "Cisco: Meraki Organization Discovery [API]" and click **[Select]**.
7. In the **Align Dynamic Application** modal, de-select the **Use Device SNMP Credential** box. Click the **Choose Credential** option that appears.
8. Select the Cisco Meraki credential you created and click **[Select]**.
9. Click **[Align Dynamic App]** to align the Dynamic Application with the Cisco Meraki virtual device.

After aligning the "Cisco: Meraki Organization Discovery [API]" Dynamic Application, your Cisco Meraki component devices will be discovered and classified.

NOTE: The **Poll Frequency** of the "Cisco: Meraki Organization Discovery [API]" Dynamic Application should be set to 5 minutes.

Assigning a Device Class to a Discovered Device

As of version 111 of the Cisco: Meraki API PowerPack, the parameters used to assign a device class to a device have been updated. The device's model name is split into two parts when SL1 assigns a device class:

- The first two characters of the model name are assigned to the Class Identifier 1 component identifier on the Class Identifier collection object. If the Cisco: Meraki device's model name starts with a "v", the next two characters after the "v" are assigned to this collection object instead.

- The rest of the characters in the model name are assigned to the Class Identifier 2 component identifier on the Class Identifier 2 collection object.

SL1 will assign a device class based on the information in the Class Identifier 1 and 2 fields:


- If the information in the Class Identifier 1 and 2 fields matches the Class Identifier 1 and 2 of a device class in the PowerPack exactly, that device class is assigned to the device.
- If the information in the Class Identifier 1 field matches, but no match occurs on the Class Identifier 2 collection object, one of 15 fallback device classes is assigned. The fallback device class matching Class Identifier 1 with the lowest weight will be assigned.
- If no match occurs in either of the Class Identifier fields, the "Cisco Systems | Meraki Device" default device class is assigned to the device.

NOTE: This process is standard SL1 functionality that is documented in the Dynamic Application Development manual .

Manually Aligning the Cisco: Meraki Organization Discovery Dynamic Application in the SL1 Classic User Interface

After creating the Cisco Meraki virtual device, you must manually align the "Cisco: Meraki Organization Discovery [API]" Dynamic Application to the Cisco Meraki virtual device.

To manually align the Cisco Meraki Dynamic Application:

1. Go to the **Device Manager** page (Devices > Device Manager).
2. Click the wrench icon () for your Cisco Meraki virtual device.
3. In the **Device Administration** panel, click the **[Collections]** tab. The **Dynamic Application Collections** page appears.
4. Click the **[Actions]** button and select *Add Dynamic Application* from the menu.
5. In the **Dynamic Application Alignment** window, from the **Dynamic Applications** field, select the "Cisco: Meraki Organization Discovery [API]" Dynamic Application.
6. In the **Credentials** field, select the Cisco Meraki credential you created.
7. Click **[Save]** to align the Dynamic Application with the Cisco Meraki virtual device.

After aligning the "Cisco: Meraki Organization Discovery [API]" Dynamic Application, your Cisco Meraki component devices will be discovered and classified.

IMPORTANT: The virtual device named "Cloud controller" must remain in SL1 in order to discover new organizations, as they are created in the Meraki account aligned to the credential on this device. This virtual device can be deleted, but no new Meraki organizations will be discovered.

NOTE: The **Poll Frequency** of the "Cisco: Meraki Organization Discovery [API]" Dynamic Application should be set to 5 minutes.



Bulk Unaligning a Dynamic Application from Devices

You can unalign a Dynamic Application from devices manually, or bulk unalign the Dynamic Application from multiple devices.

IMPORTANT: Upgrading from a prior version of the Cisco: Meraki API PowerPack to version 1.12 will align the "Cisco: Meraki Uplink Performance [API]" Dynamic Application to both Meraki networks and network devices. To avoid this, ScienceLogic recommends unaligning the "Cisco: Meraki Uplink Performance [API]" Dynamic Application from Meraki networks before upgrading to version 1.12.

CAUTION: Upgrading to version 1.12 will cause you to lose historical data for the "Cisco: Meraki Uplink Performance [API]" Dynamic Application. Additionally, Dynamic Applications aligned to "network" devices will stop collecting.


To bulk unalign a Dynamic Application from multiple devices:


1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Type "Cisco: Meraki Uplink Performance [API]" in the **Dynamic Application Name** column.
3. Click the wrench icon () and then select the **[Subscribers]** tab. The Application Subscribers page appears.
4. Select the checkbox for each device you want to apply the action to. To select all checkboxes for all devices, select the checkbox icon () at the top of the page.
5. In the **Select Action** drop-down list, select *Unalign Device and Remove Collection Data*. This option unaligns the selected device from the Dynamic Application and deletes all historical data collected by the Dynamic Application from the device. The device is no longer considered a subscriber to the Dynamic Application. If you perform this option and later want to subscribe to this Dynamic Application again, you must re-align the device with the Dynamic Application.
6. Click the **[Go]** button to apply the action to all selected devices.

Configuring Dynamic Applications to Hide Empty Rows

If you have a device that is no longer being monitored and a configuration Dynamic Application is returning empty rows in the **[Configs]** tab of that device, you can use the *Hide row* setting in the Dynamic Applications to hide those empty rows.

To do this:



1. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
2. Located the *Cisco: Meraki [API]* PowerPack and click its wrench icon ()

3. In the pane on the left, select **Dynamic Applications**.
4. For each Dynamic Application with "Configuration" in its **Type**, click its wrench icon ()
5. In the **Dynamic Applications Properties Editor**, click the **Null Row Option** dropdown and select *Hide row*.
6. Click **[Save]**.

Disabling the Encoding Fix in the Cisco: Meraki Request Manager [API] Dynamic Application

As of version 113.5 of the PowerPack, a new "encoding" method was added within the "Request Manager" snippet of the "Cisco: Meraki Request Manager [API]" Dynamic Application in order to avoid the default SL1 behavior of displaying hex code for some characters outside the ASCII character set for Meraki network and device names. Additionally, this encoding change can be toggled off for the "Cisco: Meraki Organization Discovery [API]" Dynamic Application whereas it was previously always on. The "encoding" function passed is used to translate non-ASCII characters to their approximate ASCII equivalents using the "fix_encoding" method in the silo-apps library. For any characters that cannot be converted directly by the method above, the snippet "encoding" function also allows you to specify additional replacements, as defined in the "prefix_encoding" dictionary.

The encoding fix can be disabled within the snippet of the Dynamic Application:

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Type "Cisco: Meraki Request Manager [API]" in the **Dynamic Application Name** column.
3. Click the wrench icon () and then select the **[Snippets]** tab. The **Snippet Editor & Registry** page appears.
4. Click the wrench icon () next to **Request Manager**.
5. Remove `, encoding=encoding` from the end of the snippet.
6. Click **[Save]**.

Updating the Polling Interval for the Cisco: Meraki Uplink Usage Performance [API] Dynamic Application

The Cisco: Meraki API does not currently expose uplink utilization directly. The "Cisco: Meraki Uplink Usage Performance [API]" Dynamic Application retrieves the limits set for the uplinks, reads the usage data from the "Cisco: Meraki Uplink Performance [API]" Dynamic Application, and then calculates the utilization based on the polling interval and the division of bytes over that polling interval into average kbps (kilobits per second). Due to this complex relationship, updating the polling interval of either Dynamic Application can affect the accuracy of the data in those Dynamic Applications.

If you want to adjust the polling interval of the "Cisco: Meraki Uplink Usage Performance [API]" Dynamic Application, you must:

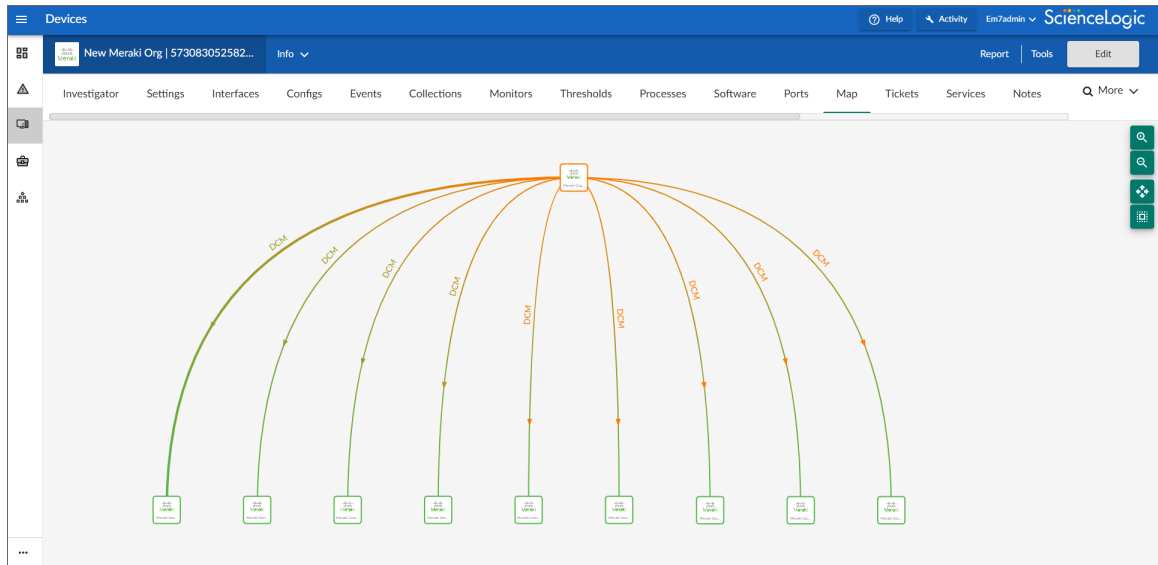
1. Update the "Cisco: Meraki Uplink Usage Performance [API]" Dynamic Application:
 - Update the polling interval of the "Cisco: Meraki Uplink Usage Performance [API]" Dynamic Application.
 - Update how many seconds the "Cisco: Meraki Uplink Usage Performance [API]" Dynamic Application is dividing the usage value by in the presentation objects in the "Cisco: Meraki Uplink Usage Performance [API]" Dynamic Application. For example, currently for the "Uplink - Download Utilization Percent" presentation object, the divisor is '37500', or $125 * \text{TIME_SPAN_IN_SEC}$. If you change the `TIME_SPAN_IN_SEC` value in the "Cisco: Meraki Request Manager [API]" Dynamic Application snippet, the divisor in the presentation object must be changed.
2. Update the "Cisco: Meraki Request Manager [API]" Dynamic Application:
 - Update the `TIME_SPAN_IN_SEC` variable in the "Cisco: Meraki Request Manager [API]" Dynamic Application Snippet. This value should match the polling interval of the "Cisco: Meraki Uplink Usage Performance [API]" Dynamic Application.
 - Add a new key:value item, for example, `"pollFrequency": TIME_SPAN_IN_SEC` in this dictionary in the "Cisco: Meraki Request Manager [API]" Dynamic Application:

```
{  
  
  "state": True,  
  
  "appGuid": "1D0E7B2EBCE646AD4E6A1CD23BAB48A7",  
  
  "endpoint": "/api/v1/organizations/{organization_  
id}/appliance/uplinks/usage/byNetwork?timespan={time_span}".format  
(organization_id=get_organization_id(), time_span=TIME_SPAN_IN_SEC),  
  
  "cacheKey": "ORGANIZATION_{root_did}_TO_APPLIANCE_UPLINK_USAGE_  
RESPONSE".format(root_did=self.did),  
  
  "pollFrequency": TIME_SPAN_IN_SEC  
}
```

Viewing Cisco Meraki Component Devices

In addition to the **Devices** page, you can view your Cisco Meraki devices in the following places in the user interface:

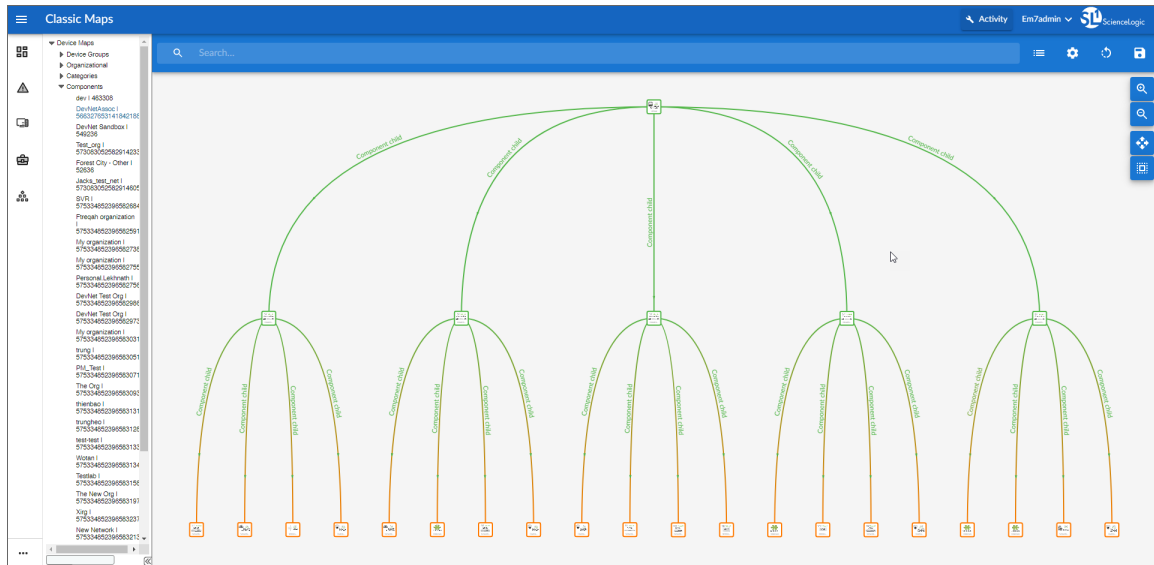
- The **Device Investigator** Map page (click **Map** in the **Device Investigator** page) displays a map of a particular device and all of the devices with which it has parent-child relationships. Double-clicking any of the listed devices reloads the page to make the selected device the primary device.



- The **Device Components** page (Devices > Device Components) displays a list of all root devices and component devices discovered by SL1. The **Device Components** page displays all root devices and component devices in an indented view, so you can easily view the hierarchy and relationships between child devices, parent devices, and root devices. To view the component devices associated with a Cisco Meraki device, find the device and click its plus icon (+).

Device Name	IP Address	Device Category	Device Class / Sub-class	DID	Organization	Current State	Collection State	Collection State
New Meraki Org 57308305258215025	--	Virtual	Cisco Systems Meraki Organization	1430	System	Major	CUG	Active
Long Island Office test	--	Network	Cisco Systems Meraki Combination Network	1579	System	Healthy	CUG	Active
LongCktest	--	Network	Cisco Systems Meraki Combination Network	1581	System	Healthy	CUG	Active
my automated network	--	Network	Cisco Systems Meraki Switch Network	1578	System	Healthy	CUG	Active
my island new automated network	--	Network	Cisco Systems Meraki Switch Network	1580	System	Healthy	CUG	Active
my new automated network	--	Network	Cisco Systems Meraki Switch Network	1577	System	Healthy	CUG	Active
my new test	--	Network	Cisco Systems Meraki Security Appliance Network	1583	System	Healthy	CUG	Active
my-emo-lad-mv	--	Network	Cisco Systems Meraki Security Appliance Network	1585	System	Healthy	CUG	Active
my test	--	Network	Cisco Systems Meraki Security Appliance Network	1584	System	Healthy	CUG	Active
my test	--	Network	Cisco Systems Meraki Switch Network	1582	System	Healthy	CUG	Active

- The **Component Map** page (Classic Maps > Device Maps > Components) allows you to view devices by root node and view the relationships between root nodes, parent components, and child components in a map. This makes it easy to visualize and manage root nodes and their components. SL1 automatically updates the **Component Map** as new component devices are discovered. The platform also updates each map with the latest status and event information. To view the map for a Cisco Meraki device, go to the **Component Map** page and select the map from the list in the left NavBar. To learn more about the **Component Map** page, see the **Maps** manual.



Creating Events from Cisco Meraki Emails

The *Cisco: Meraki [API]* PowerPack includes Event Policies that can generate events in SL1 based on emails that Cisco Meraki sends to SL1. On SL1 version 11.2 and greater, Webhooks can be configured to handle these same events.

For SL1 to process events from inbound emails, you must configure your Meraki devices to send email to SL1 using certain formatting rules.

You must then enable SL1 to generate events from those inbound Meraki emails.

If configured properly, when SL1 domain receives an email with body text that matches a Meraki network component device name and a subject that matches the regular expression (RegEx) pattern of one of the PowerPack's Event Policies, SL1 will generate an event aligned to that network component device.

NOTE: Events from email are always aligned to network devices, even when the email includes references to one or more sub-component devices below the network device.

CAUTION: The email Event Policies included in the *Cisco: Meraki [API]* PowerPack each have an expiry delay setting that specifies the amount of time after which an active event is automatically cleared from SL1 if the event has not reoccurred. However, SL1 clearing an event for reaching its expiry delay setting does not mean that the initial condition that caused the event has been resolved.

Formatting Inbound Emails

Inbound emails must meet the following requirements to be processed as events by SL1:

- The email must be sent to the following address:

```
notify@SL1-domain-name
```

Where "SL1-domain-name" is one of the fully qualified domain names of the Database Server or All-In-One Appliance that is entered in the **Authorized Email Domains** field in the **Email Settings** (System > Settings > Email) page.

- The "from" address used by the external device must be "alerts-noreply@meraki.com" for non-maintenance events, "support-noreply@meraki.com" for maintenance events, or otherwise match an address defined in the **Originator Address** field in an email redirection policy on the **Mailer Redirection** page Events > Inbound Email, or Registry > Events > Inbound Email in the SL1 classic user interface).
- The email subject line must begin with "Alert for" or "Scheduled maintenance for" and match the regular expression (RegEx) pattern of one of the Event Policies included in the *Cisco: Meraki [API] PowerPack*.
- The email body must include the name of a network device monitored by the SL1 system.

The following RegEx patterns are used:

- For scheduled maintenance emails:

```
(Scheduled maintenance for)\s  
((network\s|\d\snetworks\s|\sin\sorganization\s)"([a-zA-Z0-9_\-\.]+).*)"
```

- For all other emails:

```
(Alert for)\s*([a-zA-Z0-9_\-\.]+)\s*
```

NOTE: There must be a space between the RegEx pattern and the IP address, hostname, or device ID.

NOTE: The Event Policies included in the *Cisco: Meraki [API] PowerPack* **do not** include RegEx patterns "out of the box". Users can add or modify Event Policy RegEx patterns to best suit their needs.

NOTE: Emails that do not match the RegEx pattern of any Meraki Event Policy will generate a message in the system log. Emails that do not match the name of any component device in SL1 will not generate any events or messages.

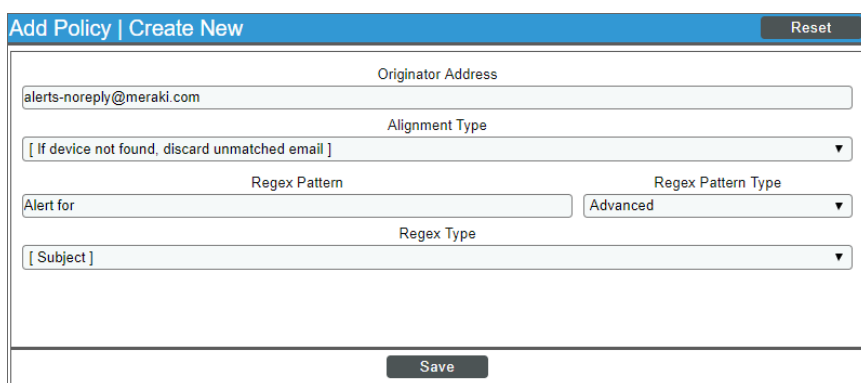
NOTE: You can specify how an Event from Email policy will match a RegEx to a device name in the **Behavior Settings** page (System > Settings > Behavior). For more information, see the *Configuring Inbound Email* manual.

Enabling Inbound Email Alerts

After you have ensured that inbound Meraki emails are formatted correctly, you must enable SL1 to generate events from the inbound Meraki emails.

To do so:

1. Go to the **Emailer Redirection** page (Events > Inbound Email, or Registry > Events > Inbound Email in the SL1 classic user interface), and then click the **[Create]** button. The **Add Policy** modal page appears.
2. Complete the following fields:



- **Originator Address.** Type "alerts-noreply@meraki.com".
- **Alignment Type.** Select *If device not found, discard unmatched email*.
- **Regex Pattern.** Type "Alert for" or "Scheduled maintenance for network".
- **Regex Pattern Type.** Select *Advanced*.
- **Regex Type.** Select *Subject*.

3. Click **[Save]**.

NOTE: For more information about generating events from inbound emails, see the *Configuring Inbound Email* manual.

Configuring Cisco: Meraki Webhooks

Please be sure to meet the following requirements before configuring webhooks for Cisco: Meraki:

- **SL1 version:** 11.2.0 or later. For details on upgrading SL1, see the appropriate SL1 Release Notes.
- The webhook handler is built to match alerts to a device based on their hostname, so you must have unique hostnames for devices across the organization.
- Message Collector with a static public IP address
- CA signed certificate for the Message Collector
- Custom webhook handler and libraries

IMPORTANT: The custom webhook handler and libraries are not included in the PowerPack. For additional information, please reach out to your Client Success Manager.

NOTE: The following steps are just one way to configure Meraki Webhooks and these steps are not the only way to configure this integration. The Cisco: Meraki [API] PowerPack is not needed to configure webhooks. For more information, please see the **Events** manual.

1. In the classic SL1 user interface, go to the **Process Manager** page (System > Manage > Applications).
2. Enable the Webhook Collector Process. For more information about this, see the **Events** manual.
3. Configure the Message Collector for Webhooks. For more information about this, see the **Events** manual.
4. Upload the CA signed certificate to the Message Collector to the path below and update the SSL configuration to use the new certificate:

```
cd /etc/nginx
```

```
sudo sed -i 's/siloss1/cert_file_name/g' /etc/nginx/conf.d/em7_
webhook_collector.conf
```



5. Go to the Collector Groups page (System > Settings > Collector Groups).
6. Align the Message Collector to the Collector Groups. For more information about this, see the **Events** manual.
7. Add a Library with Webhook Handlers. For more information about this, see the **Events** manual.

NOTE: You must write your own library or contact your Client Success Manager for details on how to get a library from ScienceLogic.

1. Go to the **Device Manager** page Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface.
2. From the **[Actions]** menu, select *Create Virtual Device*.
3. The **Create Virtual Device** modal appears.

4. Supply a value in each of the following fields:
 - **Device Name.** Name of the virtual device. Can be any combination of alphanumeric characters, up to 32 characters in length.
 - **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" poller. Select from the drop-down list of all collectors in SL1.
5. Select the **[Add]** button to save the new virtual device. Create a virtual device for each organization.

NOTE: If devices are in user disabled or maintenance mode, a webhook alert generated for those devices will get aligned to the webhook virtual device instead of the devices themselves. ScienceLogic recommends adding these webhook virtual devices in the suppression section for the webhook event policies to avoid false alerts during maintenance activities for these devices.

1. Go to the **Webhooks** page (Registry > Monitors > Webhooks)
2. Click the **[Create]** button. The **Create New Webhook** modal appears.
3. On the **Create New Webhook** modal, select the virtual device that you want to align to the new webhook receiver by clicking its device icon (.
4. On the **Create New Webhook** modal, select the `webhook_handler_example` Library by clicking its library icon (.

5. On the **Create New Webhook** modal, complete the following fields:
 - **Device**. Displays the device that you selected to align to the webhook receiver. Click **[Change Selected Device]** to select a different device.
 - **Webhook Name**. Type a unique name for the webhook receiver.
 - **Webhook URL Suffix**. Type a unique URL suffix for the webhook receiver.
 - **Available Webhook URL**. Displays the auto-generated full URL of the webhook receiver. The webhook URL consists of the IP address and port number of the Message Collector that is associated with the selected device's collector group, the static URL fragment `/api/v1/webhook`, and the webhook URL suffix, in the following format:

```
https://<IP address>:<port  
number>/api/v1/webhook/<webhook URL suffix>
```

For example: `https://10.2.20.56:8888/api/v1/webhook/test_webhook_url`

- **Library**. Displays the ScienceLogic Library that you selected to align to the webhook receiver. Click **[Change Selected Library]** to select a different ScienceLogic Library.
 - **Import Module**. Type `webhook_handler_example.meraki_example` in this field. This is the Python handler module that you want to import from the selected ScienceLogic Library.
 - **Import Handler**. Type `meraki_default_template` in this field. This is the name of the Python handler function that you want to import from the selected ScienceLogic Library.
6. Copy the **Available Webhook URL**. You will need this to configure the webhooks on the Meraki portal. Replace the `<IP address>` with the Public IP address or the DNS name of the Message Collector.
 7. Click **[Save]**.
 8. Go to the **Alerts** page on the Meraki portal. (Organization > Network-wide > Configure > Alerts)
 9. Under **Webhooks**, paste the Available Webhook URL you copied in step 18.
 10. Under **Alerts Settings**, add the Webhook Name as one of the default recipients. This step must be completed for each network under each organization.
 11. Make sure the webhook configuration synced with the Message Collector by executing the commands below in the Message Collector. This may take a while depending on the health of the system and the infrastructure:

```
silos_mysql -e 'SELECT * from master.webhook_definition;'
```

```
silos_mysql -e 'SELECT * from master.webhook_ingest;'
```

```
silos_mysql -e 'SELECT did FROM collector_state.V_device where did=virtual_  
device_id'
```

12. For examples of Meraki webhook alerts, see the Cisco Meraki documentation.

There is nothing to facilitate webhooks included in the Cisco: Meraki [API] PowerPack

Managing the Cisco: Meraki Update Switch Configuration [API] Run Book Action Policy

The "Cisco: Meraki Update Switch Configuration [API]" run book action policy is intended to be an example of how to use SL1 run book actions to automate actions within Meraki. Before being modified, this run book action policy contains code to push configuration changes to Meraki switches. Specifically, this run book action policy contains code to update the POE status of a port on a switch. This automation is intended to be easily modified to perform any Meraki API call as a result of an event occurring in SL1. This automation can be run manually for testing purposes or triggered automatically by an event that has the required information in the event message. The "Cisco: Meraki Switch Port POE has been disabled" alert and event policies are provided as examples of what events can be used to trigger this automation.

Run book automation policies:

- Cisco: Meraki Update Switch Port Config
- Cisco: Meraki Update Switch Port Config [Manual Execution]

Automation action policy:

- Cisco: Meraki Update Switch Configuration [API]

Example Alert:

- Cisco: Meraki Switch Port POE has been disabled

Example Event Policy:

- Cisco: Meraki Switch Port POE has been disabled

In an unmodified state, the "Cisco: Meraki Update Switch Configuration [API]" action policy needs 4 entities to push the configuration to the switch:

- Serial ID of the switch (This comes from the alert.)
- Port ID (This comes from the alert.)
- Switch Port Attribute (This is manually added in the "Cisco: Meraki Update Switch Configuration [API]" run book action policy.)
- Switch Port Attribute Value (This is manually added in the "Cisco: Meraki Update Switch Configuration [API]" run book action policy.)

WARNING: The "Cisco: Meraki Update Switch Configuration [API]" run book action policy and its associated automation policies are all experimental and ScienceLogic recommends caution before enabling and using them. The related "Cisco: Meraki Switch Ports Configuration [API]" and "Cisco: Meraki Switch Ports Status Configuration [API]" Dynamic Applications require a large number of API calls, which may negatively impact performance.



Executing the Cisco: Meraki Update Switch Configuration [API] Run Book Action Policy Automatically

To execute the "Cisco: Meraki Update Switch Configuration [API]" run book action policy automatically, you can use the "Cisco: Meraki Update Switch Port Config" run book automation policy.

1. Align the appropriate event created from the "Cisco: Meraki Switch Port POE has been disabled" alert to the "Cisco: Meraki Update Switch Port Config" automation policy. For more information about this, see the **Run Book Automation** manual.
2. If you would like to use this automation to change a value other than the Switch Port POE Status, configure the snippet code for the "Meraki Update Switch Configuration [API]" run book action policy. For more information about this, see the **Run Book Automation** manual.
 - Go to the `get_event_action_fields()` function and replace the `<switchPortAttribute> : <attributeValue>` with the switch port attribute that needs to be configured and its appropriate value under the `payload` attribute.
3. Enable the "Cisco: Meraki Update Switch Configuration [API]" run book action policy. For more information about this, see the **Run Book Automation** manual.
4. Enable the "Cisco: Meraki Update Switch Port Config" run book action policy. For more information about this, see the **Run Book Automation** manual.



Enabling and Configuring the Alert

The "Cisco: Meraki Switch Port POE has been disabled" alert is an example alert that is triggered when the POE status on a switch is disabled. This alert and its associated event can be used to change the PoE Status configuration from "False" to "True".

1. Go to the **Dynamic Applications Manager** page (System > Settings > Processes).
2. Type "Cisco: Meraki Switch Ports Status Configuration [API]" in the **Dynamic Application Name** column.
3. Click the wrench icon () and then select the **[Alerts]** tab. The **Alert Objects** page appears.
4. Click the wrench icon () next to the **Policy Name** in the Alert Object Registry.
5. In the **Active State** field, select *Enabled*.
6. Click **[Save]**.

Manually Executing the Cisco: Meraki Update Switch Configuration [API] Run Book Action Policy

To manually execute the "Cisco: Meraki Update Switch Configuration [API]" run book action policy, you can use the "Cisco: Meraki Update Switch Port Config [Manual Execution] [API]" run book automation policy. When using the user initiated automation policy, the Serial ID, Port ID, Switch Port Attribute, Attribute Value, and SL1 Credential ID are all added manually in the run book action policy.

1. Configure the snippet code for the "Meraki Update Switch Configuration [API]" run book action policy. For more information about this, see the **Run Book Automation** manual.
 - In the **Snippet Code** field, go to the `get_event_action_fields()` function and locate `get_user_action_fields()`.
 - Add the `<switchPortAttribute> : <attributeValue>`, the `serial`, `portId` and `credentialId`. Uncomment the `switchPortAttribute` and `portId` lines and changes the values as necessary. The key/value pair for the `switchPortAttribute` line in the payload section should follow the format provided by the Cisco Meraki documentation.
2. Go to the **Events** page.
3. Choose any event in the list and click on the message under the **Message** column. This will open the event.
4. Select the *Cisco: Meraki Update Switch Port Config [Manual Execution]* policy from the **Tools** drop-down menu. The **View Logs** link will appear.
5. Click **View Logs** to verify that the action policy was executed correctly. In the window that appears, all action policies that have been executed are listed for review.
6. Once the policy and action are enabled , go back to Device Components (Registry > Devices > Device Components).
7. Click the wrench icon () next to the switch device that triggered the alert and click the **[Logs]** tab.
8. If the run book action policy has executed correctly, an email icon () will display next to the **Count** column. Click the icon to check the execution state.

Managing the Cisco: Meraki Reboot Device Run Book Action Policy

The "Cisco: Meraki Reboot Device " run book action policy allows you to reboot a Meraki device. This action policy is disabled by default.

WARNING: The "Cisco: Meraki Reboot Device" run book action policy will allow SL1 to reboot devices. This action policy is experimental and should be turned on only by a user with extensive knowledge of the effects that these actions will have on your network and devices. ScienceLogic recommends caution when enabling this action policy in a production environment.

In order to execute this experimental action policy, you need to meet the following prerequisites:

- Enable the "Cisco: Meraki Reboot Device" run book action policy
- Enable the "Cisco: Meraki Reboot Automation [Manual Execution]" or "Cisco: Meraki Reboot Automation" run book automation policy
- Meraki credentials with the permissions to perform an http POST request

Manually Executing the Cisco: Meraki Reboot Device Run Book Action Policy

The "Cisco: Meraki Reboot Device" run book action has two policies:

- Cisco: Meraki Reboot Automation [Manual Execution]
- Cisco: Meraki Reboot Automation

To manually execute the "Cisco: Meraki Reboot Device [Manual Execution]" run book action policy:

1. Enable the "Cisco: Meraki Reboot Automation [Manual Execution]" or the "Cisco: Meraki Reboot Automation" run book automation policy depending upon your use case scenario. For more information about this, see the **Run Book Automation** manual.
2. Modify the snippet code in the "Cisco: Meraki Reboot Device" action policy to manually reboot devices. For more information about this, see the **Run Book Automation** manual.
 - In the **Snippet Code** field, go to the "Update Values Here to Run Manually" section.
 - Enter the following details:
 - `user_serial_id`, for example: `user_serial_id='QBSB-X4HM-KJVV'`
 - `user_cred_id`, for example: `user_cred_id='104'`
3. Enable the "Cisco: Meraki Reboot Device" run book action policy. For more information about this, see the **Run Book Automation** manual.
4. Go to the **Events** page.
5. Choose any Event in the list to align to the run book action and click on the message under the **Message** column. This will open the event.
 - This event will provide the serial number of the device that needs to be rebooted.
 - The run book action will do a reboot (HTTP put) on the serial ID.
6. Disable the "Cisco: Meraki Reboot Device" run book action and the automation policy "Cisco: Meraki Reboot Device [Manual Execution]" or the "Cisco: Meraki Reboot Automation".

Select the *Cisco: Meraki Reboot Automation [Manual Execution]* policy from the **Tools** drop-down menu. The **View Logs** link will appear.

7. Click **View Logs** to verify that the action policy was executed correctly. In the window that appears, all action policies that have been executed are listed for review.

Executing the Cisco: Meraki Reboot Device [API] Run Book Action Policy Automatically

To execute the "Cisco: Meraki Reboot Device [API]" run book action policy automatically, you must use the "Cisco: Meraki Reboot Device" run book automation policy.

1. In the "Cisco: Meraki Reboot Device" automation policy, select the event you would like to trigger this automation from. This event must contain the serial number for the device being rebooted in the event message. For more information about this, see the SL1: Administration and Accounts manual. For more information about this, see the **Run Book Automation** manual.
2. Enable the "Cisco: Meraki Reboot Device" run book action policy. For more information about this, see the **Run Book Automation** manual.
3. Enable the "Cisco: Meraki Reboot Automation" run book automation policy. For more information about this, see the **Run Book Automation** manual.

Using Custom Device Classes with Cisco Meraki API

If you have Cisco Meraki devices whose device classes do not match those contained in the Cisco Meraki API PowerPack, you can create and add custom device classes to the pack in order to discover them in SL1.

Creating a Custom Component Device Class

The Cisco Meraki API PowerPack includes device classes for many Cisco Meraki devices, but you can create custom device classes for devices that do not meet the criteria of the device classes in the PowerPack

To create a custom Component Device Class:

1. Go to the Device Class editor page (System > Customize > Device Classes).
2. Click **[Reset]** to clear the fields in the **Device Class Editor** pane.
3. Configure the device class as follows:
 - **Device Type.** Select "Component".
 - **Device Class.** Enter the name of the Device Class.
 - **Class Identifier 1.** Enter the first two characters of the model name in lower case letters here. For example, if the model name is "zx-d2", enter "zx".
 - **Class Identifier 2.** Enter the rest of the characters of the model name in lower case letters here. Do not begin this field with a "-" character. For example, if the model name is "zx-d2", enter "d2"
 - **Device Category.** Select the appropriate category from the drop-down list. This field specifies a logical categorization of devices by primary function, which allows SL1 to group related devices in reports and views.
 - **Root Device.** Select this checkbox if you will have additional tiers under this component device.
 - **Weight.** Select a value from the drop-down menu. A lower number should be assigned to a device class with a specific model(For ex , c9200-MXPOX) , A higher number should go to fallback or catch-it-all device classes.
 - **Description.** Enter a description for the device.
 - **Device Icon.** Select an icon that you created or select a generic icon.

4. Click **[Save]** to save your changes to the device class.



Adding Custom Device Classes to the PowerPack

NOTE: ScienceLogic does not recommend creating and adding your own device classes on your own. Failure to send custom device classes to your client success manager may affect billing.

If you have created custom device classes for your Cisco Meraki devices, you can add them to the PowerPack.

For more information on how to create device classes, see the **Device Management** manual.

To add device classes to the PowerPack:

1. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
2. Locate the *Cisco: Meraki [API]* PowerPack and click its wrench icon (.
3. From the **PowerPack Properties** page, click **Device Classes** in the Navbar on the left side of the page.
4. To add a device class, go to the **Available Device Classes** pane at the bottom of the page. Find the device class you want to include and click its lightning bolt icon (). The content will be moved to the top pane and included in the PowerPack.

NOTE: If a device is no longer collecting, check to see if the device tags have been changed and no longer match the tags in the credential for selective discovery.



Troubleshooting

The following sections describe resolutions to some issues you might encounter when monitoring Cisco: Meraki [API]:

Incorrect Calculations for Presentation Objects in the Cisco: Meraki Uplink Usage Performance [API] Dynamic Application

In version 114 of the "Cisco: Meraki [API]" PowerPack, there is a known issue where the calculations for the "Uplink - Download Utilization Percent" and "Uplink - Upload Utilization Percent" presentation objects in the "Cisco: Meraki Uplink Usage Performance [API]" Dynamic Application are incorrect and should be changed to 375 instead of 37500.

To change the calculations:

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Type "Cisco: Meraki Uplink Usage Performance [API]" in the **Dynamic Application Name** column.
3. Click the wrench icon () and then select the **[Presentations]** tab. The **Presentation Objects** page appears.
4. Locate the **Uplink - Download Utilization Percent** presentation object and click its wrench icon () .
5. In the **Formula Editor** field, change 37500 to 375.
6. Click **[Save]**.
7. Repeat steps 4-6 for the **Uplink - Upload Utilization Percent** presentation object.

Cisco: Meraki Uplink Performance [API] retrieves a NULL value

Currently, the behavior when the "Cisco: Meraki Uplink Performance [API]" Dynamic Application retrieves a value of "NULL" for the uplink is:

1. The string 'NoName' + Serial Number is added as the index for the graphs.
2. The 'lossPercent' value could be 100%, which may trigger alerts at the Network Device level in the component tree.
3. This behavior currently exists in the PowerPack, since it could not be determined what a "NULL" value for the uplink means and whether it is useful.
4. You can choose to alter your alert policies to ignore these "NULL" value uplinks.

Meraki Organizations are not modeling

In versions 113.5 and 113.6 of the Cisco: Meraki [API] PowerPack, there is a known issue that prevents Meraki Organizations from being modeled if another device in SL1 has the same name. If this situation occurs, SL1 modifies the existing device's name and does not create a new device. To workaround this issue:

1. Run the "Cisco: Meraki Organization Discovery [API]" Dynamic Application in debug mode.
2. Identify the organization name where the failure is happening on by either looking into the logs in the SL1 user interface or the `/tmp/meraki_organization_creation.log` file.
3. Locate the device with the similar name, and change the name of the device if possible.
4. Once you have changed the name of the device, SL1 will no longer try to update the existing device and will continue to create a new Meraki device.

Appendix

2

Cisco: Meraki [API] API Endpoints

Overview

This appendix describes the list of API endpoints being requested from and their matching Dynamic Applications, as well as the API Rates for those Dynamic Applications.

Cisco: Meraki [API] API Endpoints

The "Cisco: Meraki [API]" Dynamic Applications listed below make requests to the listed endpoints.

- Cisco: Meraki Organization Discovery [API]: `/api/v1/organizations`, List the organizations that the user has privileges on.
- Cisco: Meraki Network Discovery [API]: `/api/v1/<ORGANIZATION_ID>/networks`, List the networks that the user has privileges on in an organization.
- Cisco: Meraki Device Discovery [API]: `/api/v1/<ORGANIZATION_ID>/devices`, List the devices in an organization.
- Cisco: Meraki Device Configuration [API]: `/api/v1/<ORGANIZATION_ID>/devices/statuses`, List the status of every Meraki device in the organization.
- Cisco: Meraki Uplink Performance [API]: `/api/v1/<ORGANIZATION_ID>/devices/uplinksLossAndLatency`, Return the uplink loss and latency for every MX appliance in the organization from at least 2 minutes ago.
- Cisco: Meraki Organization License Configuration [API]: `/api/v1/<ORGANIZATION_ID>/licenses/overview`, Return an overview of the license state for an organization.
- Cisco: Meraki VPN Status [API]: `/api/v1/<ORGANIZATION_ID>/appliance/vpn/statuses`, Show VPN status for networks in an organization.

- Cisco: Meraki Uplink Status [API]: `/api/v1/<ORGANIZATION_ID>/appliance/uplink/statuses`, List the uplink status of every Meraki MX and Z series appliance in the organization.
- Cisco: Meraki Switch Ports Configuration [API]: `/api/v1/organizations/<ORGANIZATION_ID>/switch/ports/bySwitch`, List the switchports in an organization by switch
- Cisco: Meraki Switch Ports Status Configuration [API]: `/api/v1/devices/<DEVICE_SERIAL>/switch/ports/statuses`, Return the status for all the ports of a switch
- Cisco: Meraki Switch Port Status Performance [API]: `/api/v1/devices/<DEVICE_SERIAL>/switch/ports/statuses`, Return the status for all the ports of a switch
- Cisco: Meraki Uplink Usage Performance [API]: `api/v1/organizations/{organizationId}/appliance/uplinks/usage/byNetwork \ AND /networks/{networkId}/appliance/trafficShaping/uplinkBandwidth`, Return the sent and received bytes for each uplink of all MX and Z networks within an organization. Collects the uplink bandwidth limits, then presents the result as % utilization per uplink. If more than one device was active during the specified timespan, then the sent and received bytes will be aggregated by interface.

NOTE: The calls made to the endpoint used by the "Cisco: Meraki VPN Status [API]" Dynamic Application is only made if at least one instance of the Dynamic Application is aligned to a device. Additional instances of the Dynamic Application aligned to other devices does not add to the number of calls to the endpoint, and only one call is made per Polling Interval of the "Cisco: Meraki Request Manager [API]" Dynamic Application to get the VPN information for the entire organization.

NOTE: The calls made to the endpoints used by the "Cisco: Meraki Switch Ports Status Configuration [API]" Dynamic Application are only made if the Dynamic Application is manually aligned to a switch device.

The following endpoint is called by the "Cisco: Meraki Reboot Device" run book action policy instead of a Dynamic Application. Only one call is made per execution of the run book action policy:

- Cisco: Meraki Reboot Device: `/api/v1/devices/<SERIAL_ID>/reboot`, Reboot a device.

Cisco: Meraki [API] Dynamic Application API Rates

The "Cisco: Meraki [API]" Dynamic Applications below make a number of API calls at the rates listed.

- **Cisco: Meraki Organization Discovery [API]:** 1 API call per Account per Dynamic Application run.
- **Cisco: Meraki HTTP Stats [API]:** N/A
- **Cisco: Meraki Component Counts [API]:** N/A
- **Cisco: Meraki Network Discovery [API]:** 1 API call per Meraki Organization when the Request Manager runs. Paginates after 1000 networks.
- **Cisco: Meraki Organization License Configuration [API]:** 1 API call per Meraki Organization when the Request Manager runs. Paginates after results.

- **Cisco: Meraki Request Manager [API]:** N/A
- **Cisco: Meraki Device Discovery [API]:** 1 API call per Meraki Organization when the Request Manager runs. Paginates after 1000 devices.
- **Cisco: Meraki Network Component Counts [API]:** N/A
- **Cisco: Meraki Network Configuration [API]:** N/A
- **Cisco: Meraki Device Configuration [API]:** 1 API call per Meraki Organization when the Request Manager runs. Paginates after 1000 devices.
- **Cisco: Meraki Uplink Usage Performance [API]:** When the Request Manager runs, 1 API call per Meraki Organization for the usage, 1 API call per Network for the limits to calculate utilization. You can configure limiting the API call rate in the Request Manager snippet.
- **Cisco: Meraki Uplink Performance [API]:** 1 API call per Meraki Organization when the Request Manager runs. Paginates after 1000 devices with uplinks.
- **Cisco: Meraki Uplink Status [API]:** 1 API call per Meraki Organization when the Request Manager runs. Paginates after 1000 devices.
- **Cisco: Meraki VPN Status [API]:** 1 API call per Meraki Organization when the Request Manager runs. Paginates after 1000 devices.
- **Cisco: Meraki Switch Ports Configuration [API]:** 1 API call per Meraki Organization when the Request Manager runs. Paginates after 1000 devices.
- **Cisco: Meraki Switch Ports Status Configuration [API]:** 1 API call per device with this Dynamic Application aligned when the Request Manager runs.
- **Cisco: Meraki Switch Port Status Performance [API]:** N/A

© 2003 - 2025, ScienceLogic, Inc.

All rights reserved.

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>.



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

International: +1-703-354-1010