



---

## Monitoring Cisco Meraki (API)

Cisco: Meraki [API] PowerPack version 115

---

# Table of Contents

<b>Introduction</b>	<b>4</b>
What is Cisco Meraki?	4
What Does the Cisco: Meraki [API] PowerPack Monitor?	5
Best Practices for Monitoring Meraki with SL1	5
Installing the Cisco: Meraki [API] PowerPack	6
<b>Credentials</b>	<b>8</b>
Generating a Cisco Meraki API Key	8
Configuring a Credential for Cisco Meraki	9
Creating a Universal Credential for Cisco Meraki	10
Creating a SOAP/XML Credential for Cisco Meraki	12
Creating a SOAP/XML Credential in the SL1 Classic User Interface	17
Creating a Basic/Snippet Credential for Cisco Meraki	19
Creating a Basic/Snippet Credential in the SL1 Classic User Interface	21
Testing the Cisco Meraki API Credential	21
Testing the Cisco Meraki API Credential in the SL1 Classic User Interface	23
<b>Discovery and Configuration</b>	<b>25</b>
Cisco Meraki Guided Discovery	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	28
Viewing Cisco Meraki Component Devices	29
Configuring Dynamic Applications in the Cisco: Meraki [API] PowerPack	31
Bulk Unaligning a Dynamic Application from Devices	34
Configuring Dynamic Applications to Hide Empty Rows	34
Disabling the Encoding Fix in the Cisco: Meraki Request Manager [API] Dynamic Application	35
Updating the Polling Interval for the Cisco: Meraki Uplink Usage Performance [API] Dynamic Application	35
Creating Events from Cisco Meraki Emails	36
Formatting Inbound Emails	37
Enabling Inbound Email Alerts	38
Configuring Cisco Meraki Webhooks	38

Managing the Cisco: Meraki Update Switch Configuration [API] Run Book Action Policy .....	42
Executing the Cisco: Meraki Update Switch Configuration [API] Run Book Action Policy Automatically	43
Enabling and Configuring the Alert .....	43
Manually Executing the Cisco: Meraki Update Switch Configuration [API] Run Book Action Policy .....	44
Managing the Cisco: Meraki Reboot Device [API] Run Book Action Policy .....	44
Manually Executing the Cisco: Meraki Reboot Device Run Book Action Policy .....	45
Executing the Cisco: Meraki Reboot Device [API] Run Book Action Policy Automatically .....	46
Managing the Cisco: Meraki - Vanish Children Run Book Action Policy .....	46
Using Custom Device Classes with Cisco Meraki API .....	47
Creating a Custom Component Device Class .....	47
Adding Custom Device Classes to the PowerPack .....	48
Troubleshooting .....	48
Receiving 429 Response Codes from the Cisco Meraki API .....	48
Incorrect Calculations for Presentation Objects in the Cisco: Meraki Uplink Usage Performance [API] Dynamic Application .....	49
Cisco: Meraki Uplink Performance [API] Retrieves a NULL Value .....	49
Meraki Organizations are Not Modeling .....	50
<b>Cisco Meraki API Endpoints .....</b>	<b>51</b>
Cisco: Meraki [API] API Endpoints .....	51
Cisco: Meraki [API] Dynamic Application API Rates .....	52

---

# 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.

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>Best Practices for Monitoring Meraki with SL1</i> .....	5
<i>Installing the Cisco: Meraki [API] PowerPack</i> .....	6

<p><b>NOTE:</b> 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.</p>
---

---

### 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
- A universal credential that you can use to run a guided discovery of Cisco Meraki services
- 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.

---

## Best Practices for Monitoring Meraki with SL1

Cisco Meraki differs from other solutions and is focused on abstraction of complexities and ease of use. As a result, Meraki devices do not expose 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]" PowerPack 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. For more information, see the sections later in this manual 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. You can attempt to monitor Meraki devices via SNMP with an IP-based discovery, but the management information bases (MIB) 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, because Meraki requires DHCP to be enabled, and collection issues will occur if IP addresses are reassigned after merging the devices.

Some Dynamic Applications might 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 done only 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.

---

## 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.

**CAUTION:** If you have customized run book action/automation policies or alert policies, enabled or disabled certain Dynamic Applications, modified the "Cisco: Meraki Request Manager [API]" Dynamic Application snippet to configure API calls, or made similar changes in the "Cisco: Meraki [API]" PowerPack, ScienceLogic recommends backing up these changes prior to upgrading to version 114 or later. After upgrading, you can reimplement the backed up changes. If you continue to use duplicated or customized versions of the items mentioned above and do not upgrade to versions using Python 3, you will experience issues in SL1 12.5.0 and later, where Python 2 support is deprecated.

**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](#).

**NOTE:** For details on upgrading SL1, see the relevant [SL1 Platform Release Notes](#).

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

## Credentials

---

### Overview

The following sections describe how to configure the credentials required to monitor Cisco Meraki devices using the "Cisco: Meraki [API]" PowerPack and the Meraki API.

This chapter covers the following topics:

<i>Generating a Cisco Meraki API Key</i> .....	8
<i>Configuring a Credential for Cisco Meraki</i> .....	9
<i>Testing the Cisco Meraki API Credential</i> .....	21

---

### 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 [universal credential](#), [SOAP/XML credential](#), or [Basic/Snippet 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 "Cisco: Meraki Update Switch Configuration [API]" and "Cisco: Meraki Reboot Device [API]" run book action policies, which allow you to change configurations and reboot 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 Dynamic Component Map tree as children of the organization. Organizations and their child components can be moved between Data 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 Meraki 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:
  - **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**:
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.
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.

---

## Configuring a Credential for Cisco Meraki

To configure SL1 to monitor Cisco Meraki systems using the Meraki API, you must first create a credential. This credential allows the Dynamic Applications in the "Cisco: Meraki [API]" PowerPack to connect with the Cisco Meraki API. The "Cisco: Meraki [API]" PowerPack supports three credential types to use to connect with the Cisco Meraki API:

- [Universal](#)
- [SOAP/XML](#)
- [Basic/Snippet](#)

**NOTE:** ScienceLogic recommends configuring a universal credential to connect with the Cisco Meraki API to best meet your needs, but you can configure a SOAP/XML credential if needed. While it is supported, ScienceLogic does not recommend using a Basic/Snippet credential, which does not support selective discovery or other features that might be added in future releases of the PowerPack.

## Creating a Universal Credential for Cisco Meraki

To define a universal credential to access Cisco Meraki:

1. Go to the **Credentials** page (Manage > Credentials).
2. Click **[Create New]** and select *Create Meraki api Credential*. The **Create Credential** modal page appears.
3. Supply values in the following fields:
  - **Name.** Type a name for your 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 of 1500.
  - **Meraki API Key.** Type the [Meraki API key](#).
  - **Verify SSL.** Toggle on (blue) to verify SSL certificates for API calls made using this credential.
  - **Meraki API URL.** Enter the preferred API endpoint from the [Cisco Meraki documentation](#). Usually this is "api.meraki.com", but can be different depending on a number of factors, such as location.
  - **Port.** Keep the default value.
  - **Timeout\*.** Keep the default value.

### Use Proxy

Toggle on this field (blue) if you are using a proxy server to communicate with your Cisco Meraki account, then enter the values in the fields listed below:

- **Proxy Protocol.** Select *http* or *https* from the drop-down field.
- **Proxy IP.** Enter the hostname or the IP address associated with your device.
- **Proxy Port.** Enter the port number for the proxy server.
- **Proxy User.** Enter the username for the proxy server.
- **Proxy Password.** Enter the password for the proxy server.

### Ignore Endpoints

Toggle on this field (blue) if you want to ignore certain API endpoints. Then, toggle on (blue) each endpoint that you want to be ignored during discovery. For more information about the API endpoints used by each Dynamic Application and their API rates, see the [Cisco: Meraki \[API\] API Endpoints Appendix](#).

### **Discover Devices by Tag**

Toggle on this field (blue) if you want to discover devices by tag, then enter the values in the fields listed below.

- **Ignore Case.** Toggle on (blue) if you want SL1 to match the tag values regardless of case.
- **Tags.** Enter one or more tag values. You can include multiple tag values in a string, using comma separators and no spaces. For example: "tagvalue1,tagvalue2,tagvalue3".

**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:

- An asterisk ( \* ) will match any number of characters, including zero. For example, "a\*c" will match "ac", "abc", and "abbbbbc".
- A comma ( , ) will match exactly one character. For example, "a,c" will match "abc" and "adc", but not "ac" or "abbbbbc".

**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.

### **Credential Retry Override**

**IMPORTANT:** The fields below should be changed only for troubleshooting purposes. Changing these values can cause collections to take longer to run, which could result in missing data or early termination (SIGTERM).

- **Maximum number of retries.** Keep the default value of 3 unless you are troubleshooting issues.
- **Time between retries.** Keep the default value of 0 unless you are troubleshooting issues.

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

## Creating a SOAP/XML Credential for Cisco Meraki

If you access Cisco 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 using an SL1 system prior to version 11.1.0, you will not be able to duplicate the sample credential. 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).

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

- **Cisco: Meraki - API Example**, for users who want to connect to Meraki using a SOAP/XML credential

The screenshot shows the 'Edit Credential' dialog box with the following fields and sections:

- Name \***: Cisco: Meraki - API Example
- All Organizations**: ☒ Select the organizations the credential belongs to \*
- 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**: [Redacted]
- Proxy Hostname/IP**: optional
- Proxy User**: optional
- Proxy Password**: [Redacted]
- Embedded Password [%P]**: [Redacted]
- Embed Value [%1]**: [Redacted]
- Embed Value [%2]**: [Redacted]
- Embed Value [%3]**: [Redacted]
- Embed Value [%4]**: [Redacted]
- HTTP Headers**: X-Sample-Header:Sample Value
- CURL Options**: Add CURL Option
- Credential Tester**: Select Credential test, Select Collector, IP or Hostname to test \*, Test Credential button

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

Name \*

Cisco: Meraki - API Proxy Example

All Organizations

Select the organizations the credential belongs to \*

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

••••••••

Proxy Hostname/IP

10.0.0.0

Proxy Port

3128

Proxy User

<Proxy\_User>

Proxy Password

••••••••

Embedded Password [%P]

••••••••

Embed Value [%1]

Embed Value [%2]

Embed Value [%3]

Embed Value [%4]

HTTP Headers

proxy\_uri\_protocol:http

×

CURL Options

Add CURL Option

Credential Tester

Select Credential test

Select Collector

IP or Hostname to test \*

Test Credential

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

**Edit Credential**

Name \*  
Cisco: Meraki - API Example (Selective)

All Organizations ☒ Select the organizations the credential belongs to \* 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

Proxy Hostname/IP 0 Proxy Port

Proxy User optional Proxy Password

Embedded Password [%P]

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

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

HTTP Headers Add Header

tags:tag1.tag2 X

regex:IGNORECASE X

CURL Options Add CURL Option

**Credential Tester**

Select Credential test

Select Collector

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 Example copy**, **Cisco: Meraki - API - Proxy Example copy**, or **Cisco: Meraki - API Example (Selective) copy** appears.
3. Click the **[Actions]** icon (≡) for the credential copy and select **Edit**. The **Edit Credential** modal page appears.

4. 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.** Enter the preferred API endpoint from the [Cisco Meraki documentation](#). Usually this is "api.meraki.com", but can be different depending on a number of factors, such as location.
- **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:

- An asterisk ( \* ) will match any number of characters, including zero. For example, "a\*c" will match "ac", "abc", and "abbbbbc".
- A comma ( , ) will match exactly one character. For example, "a,c" will match "abc" and "adc", but not "ac" or "abbbbc".

**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.

- If you want to remove specific API endpoints from collection when using this credential, click **[Add a header]** for each endpoint to skip, and then type "Skip-Endpoint: <endpoint>". For example, to skip



collection of the /devices/availabilities endpoint, type "Skip-Endpoint:/devices/availabilities". The allowed endpoints to skip collection on are:

- /appliance/uplink/statuses
- /devices
- /devices/availabilities
- /devices/uplinksLossandLatency
- /licenses/overview
- /wireless/devices/connectionStats

**NOTE:** Any endpoints not on this list will not be skipped, even if entered. For more information about the API endpoints used by each Dynamic Application and their API rates, see the [Cisco: Meraki \[API\] API Endpoints Appendix](#).

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

**NOTE:** If you want 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.

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

- **Cisco: Meraki - API Example**, for users who want to connect to Meraki using a SOAP/XML credential
- **Cisco: Meraki - API - Proxy Example**, for users who connect to Meraki through a third-party proxy server
- **Cisco: Meraki - API Example (Selective)**, for users who want to discover only some selected devices based on tag values

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.
3. Enter values in the following fields:

### **Basic Settings**

- **Profile Name.** Type a new name for your Meraki credential.
- **URL.** Enter the preferred API endpoint from the [Cisco Meraki documentation](#). Usually this is "api.meraki.com", but can be different depending on a number of factors, such as location.
- **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:

- An asterisk ( \* ) will match any number of characters, including zero. For example, "a\*c" will match "ac", "abc", and "abbbbbc".
- A comma ( , ) will match exactly one character. For example, "a,c" will match "abc" and "adc", but not "ac" or "abbbbc"

**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.

- If you want to remove specific API endpoints from collection when using this credential, click **[Add a header]** for each endpoint to skip, and then type "Skip-Endpoint: <endpoint>". For example, to skip collection of the /devices/availabilities endpoint, type "Skip-Endpoint: /devices/availabilities". The allowed endpoints to skip collection on are:
  - /appliance/uplink/statuses
  - /devices
  - /devices/availabilities
  - /devices/uplinksLossandLatency
  - /licenses/overview
  - /wireless/devices/connectionStats

**NOTE:** Any endpoints not on this list will not be skipped, even if entered. For more information about the API endpoints used by each Dynamic Application and their API rates, see the [Cisco: Meraki \[API\] API Endpoints Appendix](#).

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

## Creating a Basic/Snippet Credential for Cisco Meraki

An example Basic/Snippet credential that you can edit for your own use is included in the "Cisco: Meraki [API]" PowerPack.

**NOTE:** ScienceLogic recommends configuring a universal credential to connect with the Cisco Meraki API to best meet your needs, but you can configure a SOAP/XML credential if needed. While it is supported, ScienceLogic does not recommend using a Basic/Snippet credential, which does not support selective discovery or other features that might be added in future releases of the PowerPack.

**NOTE:** If you are using an SL1 system prior to version 11.1.0, you will not be able to duplicate the sample credential. 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 Basic Example** sample credential, click its **[Actions]** icon (☰) and select **Duplicate**. A copy of the credential, called **Cisco: Meraki - API Basic Example 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**. Enter the preferred API endpoint from the [Cisco Meraki documentation](#). Usually this is "api.meraki.com", but can be different depending on a number of factors, such as location.
  - **Port**. Keep the default value.
  - **Username**. Keep the default value.
  - **Password**. Type the [Meraki API key](#).
5. 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

An example Basic/Snippet credential that you can edit for your own use is included in the "Cisco: Meraki [API]" PowerPack.

**NOTE:** ScienceLogic recommends configuring a universal credential to connect with the Cisco Meraki API to best meet your needs, but you can configure a SOAP/XML credential if needed. While it is supported, ScienceLogic does not recommend using a Basic/Snippet credential, which does not support selective discovery or other features that might be added in future releases of 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.
3. Complete the following fields:
  - **Credential Name.** Type a new name for the credential.
  - **Hostname/IP.** Enter the preferred API endpoint from the [Cisco Meraki documentation](#). Usually this is "api.meraki.com", but can be different depending on a number of factors, such as location.
  - **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]**.

---

## Testing the Cisco Meraki API Credential

The "Cisco: Meraki [API]" PowerPack includes two credential tests 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] (SOAP/XML) Credential tester** and **Cisco: Meraki [API] (Basic/Snippet) Credential tester** perform 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 want to test, click the **Actions** button (⋮) next to it and select *Edit/Test*. The **Edit Credential** modal appears:

3. In the **Credential Tester** pane on the right, fill out the following fields on this page:
  - **Select Credential Test.** Select **Cisco: Meraki [API] (SOAP/XML) Credential tester** or the **Cisco: Meraki [API] (Basic/Snippet) 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.
4. Click the **[Run Test]** button to run the credential test. The **Testing Credential** window appears.

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 two credential tests 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 SOAP/XML and Basic/Snippet credentials for monitoring the Cisco Meraki API using the Dynamic Applications in the "Cisco: Meraki [API]" PowerPack.

The **Cisco: Meraki [API] (SOAP/XML) Credential tester** and **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.

To test the Cisco Meraki credential:

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



---

# Chapter

# 3

## Discovery and Configuration

---

### 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>Cisco Meraki Guided Discovery</i> .....	25
<i>Creating a Cisco Meraki Virtual Device</i> .....	26
<i>Viewing Cisco Meraki Component Devices</i> .....	29
<i>Configuring Dynamic Applications in the Cisco: Meraki [API] PowerPack</i> .....	31
<i>Creating Events from Cisco Meraki Emails</i> .....	36
<i>Configuring Cisco Meraki Webhooks</i> .....	38
<i>Managing the Cisco: Meraki Update Switch Configuration [API] Run Book Action Policy</i> .....	42
<i>Managing the Cisco: Meraki Reboot Device [API] Run Book Action Policy</i> .....	44
<i>Managing the Cisco: Meraki - Vanish Children Run Book Action Policy</i> .....	46
<i>Using Custom Device Classes with Cisco Meraki API</i> .....	47
<i>Troubleshooting</i> .....	48


---

### Cisco Meraki Guided Discovery

If you are using the default SL1 user interface (AP2), you can use the guided discovery framework process in SL1 to guide you through a variety of existing discovery types. This process, which is also called "guided discovery", lets you choose a discovery type based on the type of devices you want to monitor. The guided discovery workflow includes a button for Cisco Meraki.

**NOTE:** You cannot perform a guided discovery in the classic SL1 user interface. Instead, you must [create a virtual device](#) representing the Meraki service and then [align the discovery Dynamic Application](#) to the virtual device.

To run a guided discovery:

1. On the **Devices** page () or the **Discovery Sessions** page (Devices > Discovery Sessions), click the **[Add Devices]** button. The **Select** page appears.
2. Select the **[Cisco Meraki]** button. Additional information about the requirements for device discovery appears in the **General Information** pane to the right.
3. Click **[Select]**. The **Credential Selection** page appears.

**NOTE:** During the guided discovery process, you cannot click **[Next]** until the required fields are filled on the page, nor can you skip to future steps. However, you can revisit previous steps that you have already completed.

4. On the **Credential Selection** page of the guided discovery process, select the Cisco Meraki [universal credential](#) that you configured, and then click **[Next]**. The **Root Device Details** page appears.
5. Complete the following fields:
  - **Root Device Name.** Type the name of the root device for the Cisco Meraki root device you want to monitor.
  - **Select the organization to add discovered devices to.** Select the name of the organization to which you want to add the discovered device.
  - **Collector Group Name.** Select an existing collector group to communicate with the discovered device. This field is required.
6. Click **[Next]**. SL1 creates the Cisco Meraki root device with the appropriate device class assigned to it and aligns the relevant Dynamic Applications. The **Final Summary** page appears.
7. Click **[Close]**.

**NOTE:** The results of a guided discovery do not display on the **Discovery Sessions** page (Devices > Discovery Sessions).

---

## Creating a Cisco Meraki Virtual Device

Because the Cisco Meraki service does not have a static IP address, you cannot discover a Meraki device using discovery unless you use guided discovery. Instead, you must create a **virtual device** that represents the Meraki service and then [align the discovery Dynamic Application](#) to the virtual device. 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.

**NOTE:** You must use this method if you are using the classic SL1 user interface. You can also use this method if you are using the default SL1 user interface (AP2) but prefer to not use [guided discovery](#).


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

1. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface).
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 in the SL1 classic user interface:

1. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface).
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:** SL1 will create a new root device for every Cisco: Meraki organization discovered. ScienceLogic recommends that you run a check for new root devices after manually aligning the Dynamic Application.

**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 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:

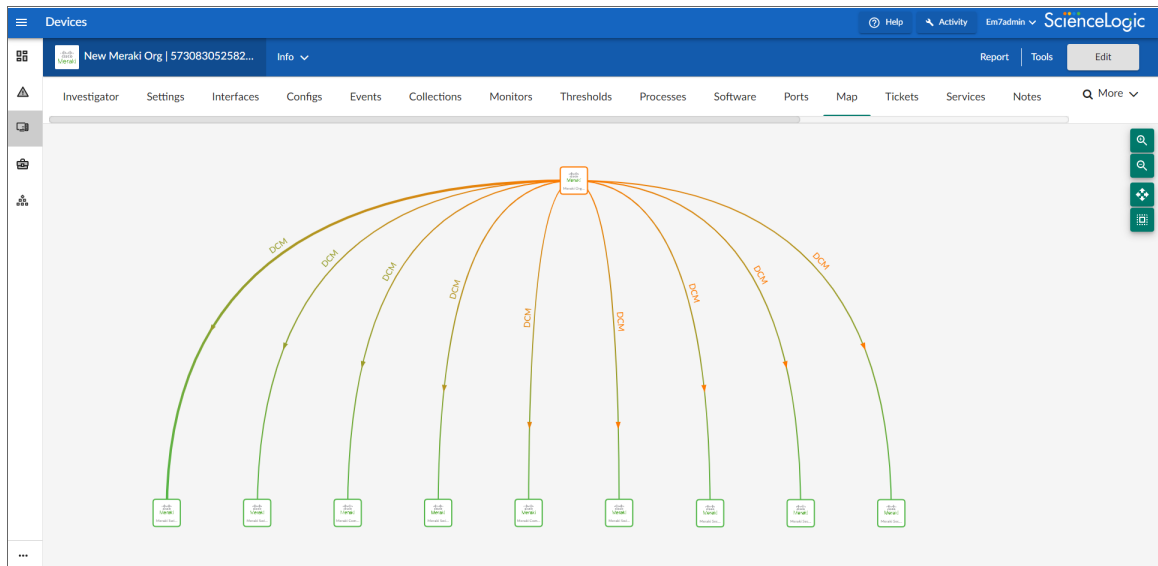
- If the information in the Class Identifier 1 and 2 fields matches the Class Identifier 1 and 2 of a device class in the PowerPack exactly, that device class is assigned to the device.
- If the information in the Class Identifier 1 field matches, but no match occurs on the Class Identifier 2 collection 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 "Configuring Collection Objects as Component Identifiers" section of the **Dynamic Application Development** manual.

# 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.

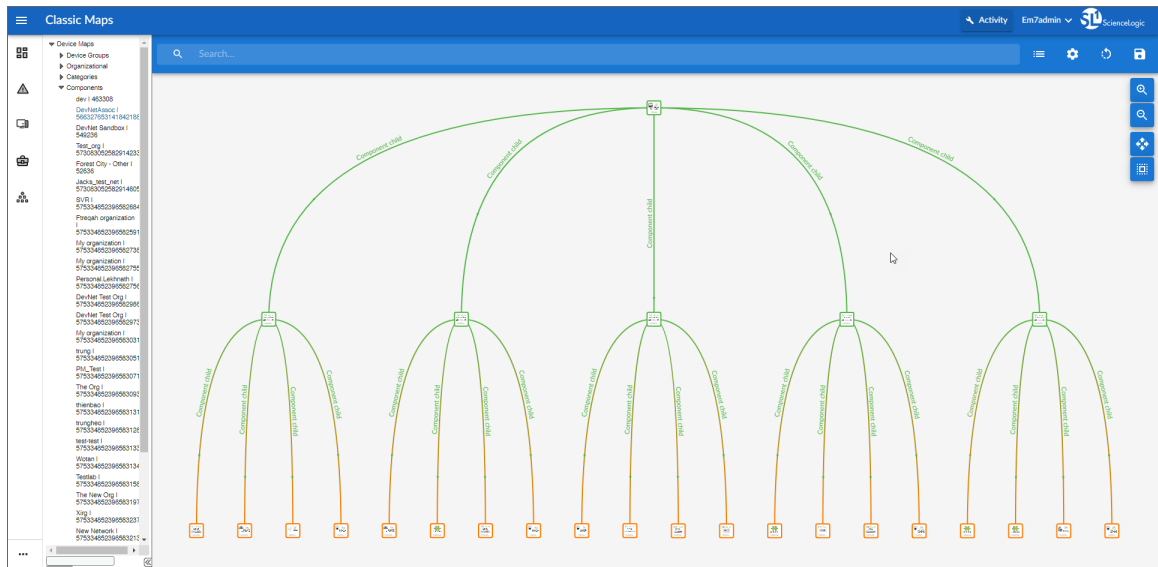


- The **Device Components** page (Devices > Device Components) displays a list of all root devices and component devices discovered by SL1. The **Device Components** page displays all root devices and component devices in an indented view, so you can easily view the hierarchy and relationships between 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 (+).

The screenshot shows the 'Device Components' page in the ScienceLogic interface. The table lists devices and their components. The first row is highlighted in orange, indicating the selected device. The table has columns for Device Name, IP Address, Device Category, Device Class / Sub-class, CID, Organization, Current State, Collection Status, and Collection State.

Device Name	IP Address	Device Category	Device Class / Sub-class	CID	Organization	Current State	Collection Status	Collection State
New Meraki Org   573083052582...	--	Virtual	Cisco Systems   Meraki Organization	1430	System	Minor	CUG	Active
Long Island Office test	--	Network	Cisco Systems   Meraki Combination Network	1570	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 brand 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
newTest	--	Network	Cisco Systems   Meraki Security Appliance Network	1583	System	Healthy	CUG	Active
10b-10b-test-new	--	Network	Cisco Systems   Meraki Security Appliance Network	1584	System	Healthy	CUG	Active
test7	--	Network	Cisco Systems   Meraki Security Appliance Network	1584	System	Healthy	CUG	Active
upnAX 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.



# Configuring Dynamic Applications in the Cisco: Meraki [API] PowerPack

The "Cisco: Meraki [API]" PowerPack supports a number of unique configuration options that allow you to monitor Cisco Meraki systems effectively.

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](#).

The devices listed below are generally expected to have data for the following Dynamic Applications if the device 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 allows only 10 API calls per organization, per requesting IP address per second.

**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.

**NOTE:** Certain Dynamic Applications can be disabled to save API calls, as indicated in the table below. Others in the "Cisco: Meraki [API]" PowerPack can be enabled or disabled by toggling the specific API calls on either a universal or SOAP/XML credential, which will affect all devices using the credential.

Additionally, Dynamic Applications that are enabled or disabled in the snippet for the Request Manager are toggled globally across all credentials.

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 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	Yes	Yes, Meraki organization	No	Meraki organization	Required to

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?
[API]					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 <b>Do Not Disable</b>	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
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



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 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
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
Cisco: Meraki Wireless Stats [API]	No	No	Yes. If aligned and enabled, the API call will be made for every network in the organization with an Access Point subcomponent.	Access points	No
Cisco: Meraki AP Utilization Performance [API]	No	No	No	Access points	No


## 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 or later 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 or later.

**CAUTION:** Upgrading to version 1.12 or later of the "Cisco: Meraki [API]" PowerPack 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 click 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 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. Locate the "Cisco: Meraki [API]" PowerPack and click its wrench icon ()
3. In the left pane, select **Dynamic Applications**.
4. Locate a Dynamic Application with "Configuration" in its **Type** column and click its wrench icon ()

5. In the **Dynamic Applications Properties Editor**, click the **Null Row Option** drop-down field and select *Hide row*.
6. Click **[Save]**.
7. Repeat steps 4-6 for each Dynamic Application in the PowerPack with "Configuration" in its **Type** column.

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

In version 113.5 of the "Cisco: Meraki [API]" PowerPack, an encoding method was added within the "Request Manager" snippet of the "Cisco: Meraki Request Manager [API]" Dynamic Application to avoid the default SL1 behavior of displaying hexadecimal 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. Previously it was 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 () for the Dynamic Application and then click 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 bits 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:
  - Change 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 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 also be changed.
2. Update the "Cisco: Meraki Request Manager [API]" Dynamic Application:
  - Change 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  
}
```

---

## 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 the 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 on the **Email Settings** page (System > Settings > Email).

- 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 classic SL1 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 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 on the **Behavior Settings** page (System > Settings > Behavior). For more information, see the "Events from Email" section in the *Configuring Inbound and Outbound 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 Cisco Meraki emails.

To do so:

1. Go to the **Emailer Redirection** page (Events > Inbound Email, or Registry > Events > Inbound Email in the classic SL1 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 section on "Events from Email" in the *Configuring Inbound and Outbound Email* manual.

---

## Configuring Cisco Meraki Webhooks

You must meet the following requirements before configuring webhooks for Cisco Meraki:

- You must be on SL1 version 11.2.0 or later. For details on upgrading SL1, see the appropriate SL1 [Release Notes](#).
- You must have unique hostnames for devices across the organization, as the webhook handler is built to match alerts to a device based on their hostname.

- Your Message Collector must have a static public IP address.
- You must have a certificate authority (CA)-signed certificate for the Message Collector.
- You must have a custom webhook handler and libraries.

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

**NOTE:** The following steps are just one way to configure Meraki webhooks; they are not the only way to configure this integration. The "Cisco: Meraki [API]" PowerPack is not needed to configure webhooks. For more information, see the "Using Webhooks to Generate Events" section in the **Events** manual.

1. Go to the **Process Manager** page (System > Settings > Admin Processes).
2. Enable the "Data Collection: Webhook Collector" process.. For more information about this, see the section on "Enabling the Webhook Collector Process in the **Events** manual.
3. Configure the Message Collector for webhooks. For more information about this, see the section on "Configuring Message Collectors for Webhooks" in 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/siloss/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 section on "Aligning a Collector Group and Devices for Webhooks" in the **Events** manual.
7. Add a ScienceLogic library with webhook handlers. For more information about this, see the section on "Adding a ScienceLogic Library with Webhook Handlers" in the **Events** manual.

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

8. Go to the **Device Manager** page Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface.
9. From the **[Actions]** menu, select *Create Virtual Device*. The **Create Virtual Device** modal appears.
10. 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.



11. Click 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.

12. Go to the **Webhooks** page (Registry > Monitors > Webhooks)
13. Click the **[Create]** button. The **Create New Webhook** modal appears.



14. 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 ()
- Select the `webhook_handler_example` Library by clicking its library icon ()
- 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.
- Copy the **Available Webhook URL**. You will need this to configure the webhooks on the Cisco Meraki portal. Replace the `<IP address>` with the Public IP address or the DNS name of the Message Collector.

15. Click **[Save]**.

16. Go to the **Alerts** page on the Meraki portal. (Organization > Network-wide > Configure > Alerts)

17. Under **Webhooks**, paste the Available Webhook URL you copied in step 15.

18. Under **Alerts Settings**, add the Webhook Name as one of the default recipients. This step must be completed for each network under each organization.

19. Make sure the webhook configuration is synced with the Message Collector by executing the commands below in the Message Collector. This might take a while depending on the health of the system and the infrastructure:

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

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

```
silo_mysql -e 'SELECT did FROM collector_state.V_device where  
did=virtual_device_id'
```

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

**NOTE:** 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 included in the PowerPack as an example of how to use SL1 run book actions to automate actions within Cisco Meraki. This run book action policy contains code to push configuration changes to Meraki switches; specifically, to update the Power over Ethernet (POE) status of a port on a switch. You can modify this automation to perform any Cisco Meraki API call as a result of an event occurring in SL1. You can run this automation manually for testing purposes, or it can be 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.

To use the "Cisco: Meraki Update Switch Configuration [API]" run book action policy, you need the following components:

- 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, you must provide the following information to the "Cisco: Meraki Update Switch Configuration [API]" action policy 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.

**CAUTION:** 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 might 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 policy to the "Cisco: Meraki Update Switch Port Config" automation policy. For more information, see the "Creating an Automation Policy" section in the **Run Book Automation** manual.
2. To change a value, configure the snippet code for the "Meraki Update Switch Configuration [API]" run book action policy. For more information, see the "Creating an Action Policy that Executes a Snippet" section in the **Run Book Automation** manual.
  - In the **Snippet Code** field, 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.
  - To specify a serial and port combination, go to the `get_event_action_fields()` function and replace the `serial` and `port_id` values.
3. Enable the "Cisco: Meraki Update Switch Configuration [API]" run book action policy. For more information, see the "Creating an Action Policy" section in the **Run Book Automation** manual.
4. Enable the "Cisco: Meraki Update Switch Port Config" run book action policy. For more information, see the "Creating an Automation Policy" section in the **Run Book Automation** manual.
5. Enable the event policy that you aligned in step 1. For more information, see the "Defining an Event Policy" section in the **Events** 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 > Manage > Dynamic Applications or System > Manage > Applications in the classic SL1 user interface).
2. Type "Cisco: Meraki Switch Ports Configuration [API]" in the **Dynamic Application Name** column.
3. Click the wrench icon () for the Dynamic Application and then click 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 "Creating an Action Policy that Executes a Snippet" section in the **Run Book Automation** manual.
  - In the **Snippet Code** field, go to the `get_user_actions_fields()` function.
  - Add the `<switchPortAttribute> : <attributeValue>`, the `serialId`, `portId` and `credentialId`.
  - Uncomment the `switchPortAttribute` and `portId` lines and change 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. Enable the "Cisco: Meraki Update Switch Configuration [API]" run book action policy. For more information, see the "Creating an Action Policy" section in the **Run Book Automation** manual.
3. Enable the "Cisco: Meraki Update Switch Port Config" run book action policy. For more information, see the "Creating an Automation Policy" section in the **Run Book Automation** manual.

---

## Managing the Cisco: Meraki Reboot Device [API] Run Book Action Policy

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

**WARNING:** The "Cisco: Meraki Reboot Device [API]" run book action policy allows 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.

To execute this experimental action policy, you must first:

- Enable the "Cisco: Meraki Reboot Device [API]" run book action policy
- Enable the "Cisco: Meraki Reboot Device [Manual Execution]" or "Cisco: Meraki Reboot Device" run book automation policy
- Have Cisco 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[API]" 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]" run book automation policy. For more information about this, see the "Creating an Automation Policy" section in the **Run Book Automation** manual.
2. Modify the snippet code in the "Cisco: Meraki Reboot Device [API]" action policy to manually reboot devices. For more information about this, see the "Creating an Action Policy" section in 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 [API]" run book action policy. For more information about this, see the "Creating an Action Policy" section in 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 to open the event.
  - This event provides the serial ID of the device that needs to be rebooted.
  - The run book action will do a reboot (HTTP PUT) on the serial ID.

6. Click the **Tools** drop-down menu and select the *Cisco: Meraki Reboot Automation [Manual Execution]* policy. The **View Logs** link appears.
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.
8. Disable the "Cisco: Meraki Reboot Device [API]" run book action and the automation policy "Cisco: Meraki Reboot Device [Manual Execution]".

## 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 ID for the device being rebooted in the event message. For more information about this, see the "Creating an Automation Policy" section in the **Run Book Automation** manual.
  - Use the following format for the event message: `serial_id:<serial number>`. For example: `Meraki: POE is disabled, serial_id:Q4AA-G999-2BBB`.
2. Enable the "Cisco: Meraki Reboot Device" run book automation policy. For more information about this, see the "Creating an Automation Policy" section in the **Run Book Automation** manual.
3. Enable the "Cisco: Meraki Reboot Automation [API]" run book action policy. For more information about this, see the "Creating an Action Policy" section in the **Run Book Automation** manual.

---

## Managing the Cisco: Meraki - Vanish Children Run Book Action Policy

The "Cisco: Meraki - Vanish Children" run book action policy and its associated run book automation policy ("Cisco: Meraki - Vanish Device") are included in the "Cisco: Meraki [API]" PowerPack to provide an option for overriding global vanish timers for devices that do not match the tags provided in the "Cisco: Meraki - API (Selective)" credential for use with selective discovery.

**NOTE:** The "Cisco: Meraki - Vanish Children" run book action policy and "Cisco: Meraki - Vanish Device" run book automation policy are disabled by default and might be removed in a future release of the "Cisco: Meraki [API]" PowerPack.

If you want to override global vanish timers for devices, you must enable the "Cisco: Meraki - Vanish Children" run book action policy and "Cisco: Meraki - Vanish Device" run book automation policy. For more information about this, see the "Creating an Automation Policy" section in 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 PowerPack 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 lowercase letters. 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 lowercase 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 example, c9200-MXPOX). A higher number should be assigned to fallback or catch-all device classes.
  - **Description.** Enter a description for the device class.
  - **Device Icon.** Select an icon that you created or a generic icon for the device class.
4. Click **[Save]** to save your changes to the device class.

The screenshot shows the "Device Class Editor" interface with the following fields and values:



- Device Type:** Component (dropdown)
- Root Device:** ☐
- Device Class Tier:** 3 (dropdown)
- Weight:** 3 (dropdown)
- Device Class:** Cisco Component Device Class
- Class Identifier 1:** CX
- Class Identifier 2:** 9200
- Description:** (empty text field)
- Device Icon:** [\_generic\_unknown.png] (dropdown)
- Device Category:** Wireless Access Point (dropdown)
- Dynamic App Alignment:** (none) (dropdown)
- Device Dashboard:** [None] (dropdown)
- Buttons:** Icons, Reset, Guide (top right); Save (bottom right)

## Adding Custom Device Classes to the PowerPack

**NOTE:** ScienceLogic does not recommend creating and adding custom device classes without consulting your customer success manager to determine how these device classes might affect billing.

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

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.

---

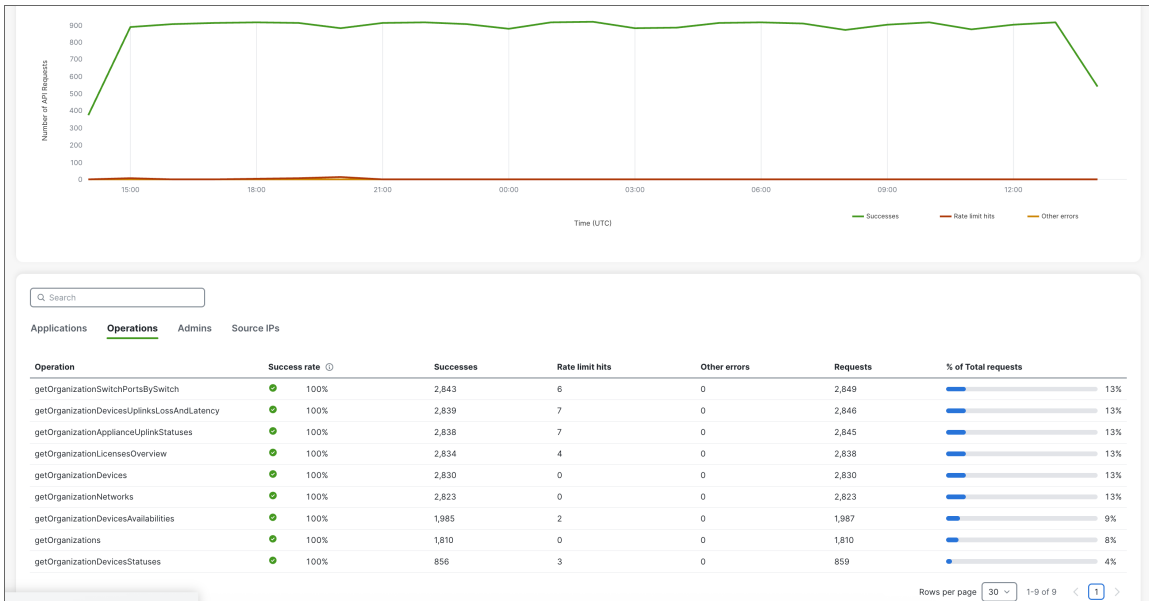
## Troubleshooting

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

### Receiving 429 Response Codes from the Cisco Meraki API

Cisco Meraki uses rate limiting to define how many API calls an API client can make in a set time frame. They limit API calls per organization, per source IP per organization, and per source IP across organizations, among others. You might receive 429 response codes from the Cisco Meraki API if the rate limit has been exceeded. This is normal behavior, and you do not necessarily need to be concerned if you receive the 429 response code. ScienceLogic recommends referencing the Cisco Meraki dashboard (on their website) to get insight into how many API calls SL1 and other tools are using, and which endpoints they are hitting.





## 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 (🔧) for the Dynamic Application and then click 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:

- The string 'NoName' + Serial Number is added as the index for the graphs.
- The 'lossPercent' value could be 100%, which may trigger alerts at the network device level in the component tree.

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.

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 work around 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 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

# 4

## Cisco Meraki API Endpoints

---

### Overview

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

---

### Cisco: Meraki [API] API Endpoints

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

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

- **Cisco: Meraki Uplink Status [API].** `/api/v1/organizations/<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 switch ports 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/<ORGANIZATION_ID>/appliance/uplinks/usage/byNetwork` AND `api/v1/networks/<NETWORK_ID>/appliance/trafficShaping/uplinkBandwidth`, return the sent and received bytes for each uplink of all Meraki MX and Z series 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 time span, then the sent and received bytes will be aggregated by interface.
- **Cisco: Meraki Wireless Stats [API].** `/api/v1/networks/<NETWORK_ID>/wireless/devices/connectionStats?timespan=<FREQUENCY_OF_WIRELESS_STATS_DYNAMIC_APP>`, which collects aggregated connectivity info for an individual network, grouped by node.
- **Cisco: Meraki AP Utilization Performance [API].** `/api/v1/organizations/{organization_id}/wireless/devices/channelUtilization/byDevice`, which returns the average channel utilization for all bands in a network, split by AP.

**NOTE:** The calls made to the endpoint used by the "Cisco: Meraki VPN Status [API]" Dynamic Application are made only if at least one instance of the Dynamic Application is aligned to a device. Having additional instances of the Dynamic Application aligned to other devices does not increase 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 1,000 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 1,000 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 1,000 devices.
- **Cisco: Meraki Uplink Usage Performance [API]**. When the Request Manager runs, 1 API call per Meraki organization for the usage and 1 API call per network for the limits to calculate utilization. You can limit 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 1,000 devices with uplinks.
- **Cisco: Meraki Uplink Status [API]**. 1 API call per Meraki organization when the Request Manager runs. Paginates after 1,000 devices.
- **Cisco: Meraki VPN Status [API]**. 1 API call per Meraki organization when the Request Manager runs. Paginates after 1,000 devices.
- **Cisco: Meraki Switch Ports Configuration [API]**. 1 API call per Meraki organization when the Request Manager runs. Paginates after 1,000 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
- **Cisco: Meraki Wireless Stats [API]**. 1 API call for each Meraki network with one or more access point subcomponents, when the "Cisco: Meraki Wireless Stats [API]" Dynamic Application is enabled and aligned to at least one access point device.

© 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](mailto:legal@sciencelogic.com). For more information, see <https://sciencelogic.com/company/legal>.



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

International: +1-703-354-1010