



Appendix B: Troubleshooting Cisco Meraki SD-WAN Issues

This section describes problems, possible causes, and recommended actions that you may encounter in Cisco Meraki SD-WAN deployments.

- [Handling Meraki Rate Limiting Issue on MSX, on page 1](#)
- [Checking Meraki Beat, on page 2](#)
- [Checking Device Status, on page 2](#)
- [Checking Device Connections, on page 6](#)

Handling Meraki Rate Limiting Issue on MSX

Error Message

Http Error 429: This error code indicates that you have submitted more than 5 API calls in one second to the Meraki system, triggering the rate limit. Meraki Dashboard API has a rate-limiting of 5 API calls per second per organization.

Solution

Modify the rate-limiting parameters through consul. Before you modify, make sure you have API access keys for Meraki. For more information, see [Managing Meraki Traffic Class Access for Tenants](#).

The following are the parameters that need to be modified:

- **meraki.ratelimit.max.attempts:** Use this parameter to configure the maximum attempts allowed for Meraki. Use the following command to configure this parameter. Run this curl command on the master Kubernetes node.

```
curl -s -k -X 'x-consul-token: <consul token>' PUT -H "Content-Type: application/json"
-d "<Number of retries>"
https://consul.service.consul:8500/v1/kv/userconfiguration/sdwanservice/meraki.ratelimit.max.attempts
| python -mjson.tool
```

- **meraki.ratelimit.backoff:** Use this parameter to retry the Meraki request after a specified backoff time. Use the following command to configure this parameter. Run this curl command on the master Kubernetes node.

```
curl -s -k -X 'x-consul-token:<consul-token>' PUT -H "Content-Type: application/json"
```

```
-d "<Backoff in milliseconds>"
https://consul.service.consul:8500/v1/kv/userconfiguration/sdwanservice/meraki.ratelimit.backoff
| python -mjson.tool
```



Note Replace {consul-token} with your actual consul token value from the passwords.yml file.

Checking Meraki Beat

Use the following endpoint to check if Meraki beat is up and running:

<IP_address>3441/admin/health

Where:

<IP_address> is the IP address of the Meraki system.

Response:

```
{"description": " metrics collector", "status": "UP"}description
```

If status is not **Up**, use the instructions in the deployment log to bring Meraki Beat up and running in kubernetes node. Check deployment logs for more information.

Checking Device Status

Checking Meraki Device Health

Use the following POST API to return all the query templates related to devices.

POST /api/v2/querytemplates/device

Response:

```
{
  "success": true,
  "responseObject": [
    {
      "type": "*",
      "profile": "*",
      "specificType": "*",
      "templateName": "query-ping-availability",
      "queryTemplate": "{\"query\": {\"bool\": {\"filter\": [{\"term\": {\"deviceId\": \"value\": \"{{deviceId}}\", \"range\": {\"@timestamp\": {\"gte\": \"{{timestamp_gte}}\", \"lte\": \"{{timestamp_lte}}\"}}]}}, \"aggs\": {\"{{statusInterval}}\": {\"date_histogram\": {\"field\": \"@timestamp\", \"interval\": \"{{statusInterval}}\", \"aggs\": {\"status\": {\"bucket_script\": {\"buckets_path\": {\"tot\": \"_count\", \"success\": \"count_success\", \"script\": \"def threshold = 0.5; if (params.success / params.tot > threshold) {return 1} else {return 0}\"}}, \"count_success\": {\"sum\": {\"script\": \"return doc['up'].value == true ? 1 : 0\"}}}}}, \"size\": 0}}\",
      \"indices\": [
        \"heartbeat-*\"
      ]
    },
    {
      "type": "*",
      "profile": "*",
      "specificType": "*",
      "templateName": "query-snmpp-availability",
```

```

    "queryTemplate": "{\"query\": {\"bool\": {\"filter\": [{\"term\": {\"deviceId\": {\"value\": \"{{deviceId}}\"}}, {\"range\": {\"@timestamp\": {\"gte\": \"{{timestamp_gte}}\", \"lte\": \"{{timestamp_lte}}\"}}]}]}, \"aggs\": {\"{{statusInterval}}\": {\"date_histogram\": {\"field\": \"@timestamp\", \"interval\": \"{{statusInterval}}\", \"aggs\": {\"status\": {\"bucket_script\": {\"buckets_path\": {\"fail\": \"count_fail\", \"tot\": \"_count\"}, \"script\": \"def threshold = 0.5; if ((params.tot - params.fail) / params.tot > threshold) {return 1} else {return 0}\"}}, \"count_fail\": {\"value_count\": {\"field\": \"Failed\"}}]}]}, \"size\": 0}\"",
    "indices": [
      "snmpbeat-*"
    ]
  },
  {
    "type": "CPE",
    "profile": "sdwan",
    "specificType": "MERAKEI",
    "templateName": "meraki-device-status",
    "queryTemplate": "{\"sort\": [{\"@timestamp\": {\"order\": \"desc\"}}], \"query\": {\"bool\": {\"filter\": [{\"term\": {\"deviceId\": {\"value\": \"{{deviceId}}\"}}, {\"range\": {\"@timestamp\": {\"gte\": \"{{timestamp_gte}}\", \"lte\": \"{{timestamp_lte}}\"}}]}]}, {\"bool\": {\"must\": [{\"exists\": {\"field\": \"DeviceHealth\"}}]}]}]}, \"aggs\": {\"{{statusInterval}}\": {\"date_histogram\": {\"field\": \"@timestamp\", \"interval\": \"{{statusInterval}}\", \"aggs\": {\"status\": {\"bucket_script\": {\"buckets_path\": {\"total\": \"_count\", \"deviceHealthSum\": \"deviceHealthSum\"}, \"gap_policy\": \"insert_zeros\", \"script\": \"(params.deviceHealthSum / params.total >= 0.5) ? 1 : 0\"}}, \"deviceHealthSum\": {\"sum\": {\"script\": \"return doc['DeviceHealth.status'].value == 'offline' ? 0 : doc['DeviceHealth.status'].value == 'online' ? 1 : 'undefined'\"}}]}]}, \"size\": 1}\"",
    "indices": [
      "merakibeat-*"
    ]
  }
],
"command": "Get all device health query templates",
"parms": {},
"httpStatus": "OK",
"message": "Get all device health query templates",
"errors": [],
"throwable": null
}

```

Problem: If meraki-device-status does not exist in the response, it means there were issues in deployment and query templates were not pushed properly.

Solution: Use the following API:

POST [api/v2/querytemplates/device](#)

Response:

```

{
  "type": "CPE",
  "profile": "sdwan",
  "specificType": "MERAKEI",
  "templateName": "meraki-device-status",
  "queryTemplate": "{\"sort\": [{\"@timestamp\": {\"order\": \"desc\"}}], \"query\": {\"bool\": {\"filter\": [{\"term\": {\"deviceId\": {\"value\": \"{{deviceId}}\"}}, {\"range\": {\"@timestamp\": {\"gte\": \"{{timestamp_gte}}\", \"lte\": \"{{timestamp_lte}}\"}}]}]}, {\"bool\": {\"must\": [{\"exists\": {\"field\": \"DeviceHealth\"}}]}]}]}, \"aggs\": {\"{{statusInterval}}\": {\"date_histogram\": {\"field\": \"@timestamp\", \"interval\": \"{{statusInterval}}\", \"aggs\": {\"status\": {\"bucket_script\": {\"buckets_path\": {\"total\": \"_count\", \"deviceHealthSum\": \"deviceHealthSum\"}, \"gap_policy\": \"insert_zeros\", \"script\": \"(params.deviceHealthSum / params.total >= 0.5) ? 1 : 0\"}}, \"deviceHealthSum\": {\"sum\": {\"script\": \"return doc['DeviceHealth.status'].value == 'offline' ? 0 : doc['DeviceHealth.status'].value == 'online' ? 1 : 'undefined'\"}}]}]}, \"size\": 1}\"",

```

```

    "indices": [
      "merakibeat-*"
    ]
  }
}

```

Checking Device Status for a Specific Service ID

Use the following end point to get the statuses of the devices for a specific service ID:

GET /api/v1/status/service/{serviceId}/devices

Response:

```

{
  "statusData": [
    {
      "id": "<Device ID 1>",
      "locationId": null,
      "name": "127.0.0.1",
      "operationalState": "up",
      "parentId": null,
      "topLevelServiceId": "<Service ID>",
      "type": "CPE"
    },
    {
      "id": "<Device ID 2>",
      "locationId": null,
      "name": "127.0.0.1",
      "operationalState": "up",
      "parentId": null,
      "topLevelServiceId": "<Service ID>",
      "type": "CPE"
    },
    {
      "id": "<Device ID 3>",
      "locationId": null,
      "name": "127.0.0.1",
      "operationalState": "down",
      "parentId": null,
      "topLevelServiceId": "<Service ID>",
      "type": "CPE"
    }
  ]
}

```

Problem: Device ID is returned empty, which indicates issue in metric collection.

Solution: In this case, run the following query for each device on Elastic Search:

```

{
  "query": {
    "bool": {
      "filter": [
        {
          "term": {
            "deviceId": {
              "value": "{{deviceId}}"
            }
          }
        },
        {
          "range": {
            "@timestamp": {
              "gte": "{{timestamp_gte}}",
              "lte": "{{timestamp_lte}}"
            }
          }
        }
      ]
    }
  }
}

```

```

    }
  },
  {
    "bool": {
      "must": [
        {
          "exists": {
            "field": "DeviceHealth"
          }
        }
      ]
    }
  }
]
},
"sort": [
  {
    "@timestamp": {
      "order": "desc"
    }
  }
],
"size": 1,
"aggs": {
  "{{statusInterval}}": {
    "date_histogram": {
      "field": "@timestamp",
      "interval": "{{statusInterval}}"
    },
    "aggs": {
      "deviceHealthSum": {
        "sum": {
          "script": "return doc['DeviceHealth.status'].value == 'offline' ? 0 :
doc['DeviceHealth.status'].value == 'online' ? 1 : 'undefined'"
        }
      },
      "status": {
        "bucket_script": {
          "buckets_path": {
            "deviceHealthSum": "deviceHealthSum",
            "total": "_count"
          },
          "gap_policy": "insert_zeros",
          "script": "(params.deviceHealthSum / params.total >= 0.5) ? 1 0"
        }
      }
    }
  }
}
}
} :[..

```

In the above query, provide the values for the following:

- timestamp_gte
- timestamp_lte
- statusInterval

Response:

```

"": {
  "": 1
}

```

0 Down
1 Up

Checking Device Connections

Use the following endpoint to get all device connections:

GET /manage/api/v2/devices/connections

Response:

```
{
  "success": true,
  "command": "getAllDeviceConnections",
  "params": {
    "serviceInstanceId": null
  },
  "message": "getAllDeviceConnections succeeded",
  "responseObject": [
    {
      "deviceInstanceId": "<Device ID1>",
      "serviceInstanceId": "<Service ID1>",
      "tenantId": null,
      "name": null,
      "profile": "sdwan",
      "type": "CPE",
      "specificType": "MERAKI",
      "category": "CPE",
      "hostName": null,
      "ipAddress": "127.0.0.1",
      "serialKey": "<Device Serial Key>",
      "createdOn": "2020-02-18T10:33:44.849412",
      "createdBy": "operatorapi181027585e8c87a9",
      "modifiedOn": "2020-02-18T10:33:44.849412",
      "modifiedBy": "operatorapi181027585e8c87a9"
    },
    {
      "deviceInstanceId": "<Device ID2>",
      "serviceInstanceId": "Service ID2",
      "tenantId": null,
      "name": "aw9DpjAZShwiRwUrNbuMztji",
      "profile": "vbranch",
      "type": "CPE",
      "specificType": "ENCS",
      "category": "CPE",
      "hostName": "aw9DpjAZShwiRwUrNbuMztji",
      "ipAddress": "127.0.0.1",
      "serialKey": "<Device Serial Key>",
      "createdOn": "2020-02-18T10:34:17.223363",
      "createdBy": "system",
      "modifiedOn": "2020-02-18T10:34:17.223363",
      "modifiedBy": "system"
    }
  ],
  "httpStatus": "OK"
}
```

Make sure the device connection has been created for the device with the serial number.

Problem: Device connections are not returned in the response.

Solution: Create a device connection using the following API :

POST /api/v2/devices/connections

Response:

```
{
  "": "CPE",category
  "": "CPE-<UUID>",deviceInstanceId
  "": null,hostName
  "": "127.0.0.1",ipAddress
  "": null,name
  "": " ",profilesDwan
  "": "<Serial_Key>",serialKey
  "": "<Service_instance_ID>",serviceInstanceId
  "": " MERAKI",specificType
  "": " CPE"type
}
```

The above endpoint response returns an entry for the device instance ID.

