



Cisco Crosswork Network Controller 6.0 Service Health Monitoring

First Published: 2023-08-28

Last Modified: 2024-05-14

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2023 Cisco Systems, Inc. All rights reserved.



CONTENTS

| | | |
|------------------|---|----------|
| CHAPTER 1 | About Cisco Crosswork Service Health | 1 |
| | Audience | 1 |
| | Overview of Cisco Crosswork Service Health | 1 |
| | Crosswork Service Health APIs | 2 |

| | | |
|------------------|---|----------|
| CHAPTER 2 | Get Started | 3 |
| | Before You Begin | 3 |
| | Getting Started | 4 |
| | Service Health Workflows | 5 |
| | Workflow: Manage Stored Data | 6 |
| | Workflow: Analyze the Cause of Service Degradation | 6 |
| | Workflow: View Performance Metrics of TE Policies using SR-PM | 7 |
| | Workflow: Monitor Service Health using Accedian Skylight | 8 |
| | Workflow: Customize Heuristic Packages | 9 |

| | | |
|------------------|--|-----------|
| CHAPTER 3 | Monitor Service Health | 11 |
| | Start Service Health monitoring | 11 |
| | Adjust Monitoring Settings | 14 |
| | Edit Existing Monitoring Settings | 14 |
| | Pause and Resume Service Health Monitoring | 15 |
| | Stop Service Health Monitoring | 16 |
| | Enable SR PM Monitoring for TE Policies | 17 |
| | Enable SR-PM Metrics Collection | 18 |
| | View SR-MPLS Policy Performance Metrics | 18 |
| | View RSVP-TE Policy Performance Metrics | 19 |
| | Monitor Service Health using Accedian Skylight | 20 |

Add Accedian Skylight as a Provider 21

View Accedian Skylight Probe Session Details 22

View Historical Data from Accedian Skylight Probe Sessions 26

Known Issues and Limitations with Accedian Skylight 27

CHAPTER 4 Analyze Degraded Services 29

View Monitored Services 29

View Monitoring Status of a Service 30

Identify Active Symptoms and Root Causes of a Degraded Service 32

About Assurance Graph 34

Identify Root Causes Using Assurance Graph 35

Identify Root Causes Using Last 24Hr Metrics 38

View the Devices Participating in the Service 41

View Collection Jobs 42

CHAPTER 5 Configure Additional Storage 45

Configure Additional External Storage 45

CHAPTER 6 Customize Heuristic Packages 49

About Heuristic Packages 49

Build a Custom Heuristic Package 51

Import Custom Heuristic Packages 53

APPENDIX A Reference - Basic Monitoring and Advanced Monitoring Rules 55

Basic and Advanced Monitoring Rules 55

Example 69

APPENDIX B Reference - Supported Subservices 79



CHAPTER 1

About Cisco Crosswork Service Health

This section explains the following topics:

- [Audience, on page 1](#)
- [Overview of Cisco Crosswork Service Health, on page 1](#)
- [Crosswork Service Health APIs, on page 2](#)

Audience

This guide is for experienced network administrators who want to use Cisco Crosswork Service Health in their network. This guide assumes that you are familiar with the following topics:

- Cisco Crosswork Infrastructure and installing Crosswork applications. For more information, see the [Cisco Crosswork Network Controller Installation Guide](#).
- Provisioning L2VPN and L3VPN services
- Networking technologies and protocols (BGP-LS, IGP (OSPF and IS-IS), PCEP, model-driven telemetry)
- Traffic Engineering (TE) tunnels:
 - RSVP-TE tunnel provisioning
 - Segment Routing Traffic Engineering (SR-TE) policy provisioning

Overview of Cisco Crosswork Service Health

Cisco Crosswork Service Health is a component of Cisco Crosswork Network Controller (Advantage package).

The application provides a ready-to-use solution supporting the following use cases:

- Monitoring the health of:
 - Point-to-point L2VPN services
 - Multipoint L2VPN (EVPN E-LAN and E-Tree L2VPN EVPN) services
 - L3VPN services
- Analysis and troubleshooting of degraded services

- Visualize the health status of a service and view its logical health dependency tree
- Extensible to add service monitoring capabilities to address specific needs

For more information on all Crosswork Network Controller solution components, see the [Crosswork Network Controller Solution Workflow Guide](#).

Crosswork Service Health APIs

Advanced users can integrate other Crosswork applications and third-party applications with Crosswork Service Health functions by using application programming interfaces (APIs) delivering new capabilities into their network operations.

For more information, see the [Cisco Crosswork Network Automation API Documentation](#) on Cisco DevNet.



CHAPTER 2

Get Started

This section explains the following topics:

- [Before You Begin, on page 3](#)
- [Getting Started, on page 4](#)
- [Service Health Workflows, on page 5](#)

Before You Begin

Before you begin using Crosswork Service Health, you are recommended to be familiar with the following concepts and complete any planning and information-gathering steps:

- Crosswork Service Health offers two levels of service monitoring - Basic and Advanced.
 - **Basic Monitoring:** This type of monitoring offers the option of adding up to 52,000 services and provides limited sub-service metrics, resulting in lower overall CPU usage. Additionally, the graphic map renderings are smaller compared to more detailed monitoring.
 - **Advanced Monitoring:** This monitoring approach allows for the addition of up to 2,000 services, resulting in higher CPU consumption, an increased number of sub-service metrics, and larger graphic map renderings.
- You can monitor upto a total of 52000 services in total, that is 52000 services using Basic monitoring only or 50000 services using Basic monitoring and an additional 2000 services using Advanced monitoring.
- Service Health uses Heuristic packages to monitor the health of the services. A Heuristic package contains what to monitor, how to compute the monitored metrics, and symptoms associated with service health degradation. The overall health of the service is determined by applying the rules from the Heuristic Package.
- The default set of Heuristic packages provided with Service Health are called system packages and cannot be modified. Based on the rules that are defined in the system packages, Service Health uses various testing probes such as IP SLA, Y.1731, TWAMP, SR-PM, Accedian Skylight, or telemetry data to analyze the health and determine if the service meets the Service Level Agreement (SLA).

You can create a custom Heuristic package by exporting an existing package, modifying it, and then importing it back. See [About Heuristic Packages, on page 49](#).
- Service Health works in environments with either standalone NSO or NSO deployed in the LSA configuration.

- Extended CLI support using Service Health system device packages allows for more comprehensive service monitoring capabilities. These packages are capable of deriving exact sensor paths for metric health calculation, and can be installed as a bundle. Engage with your account team for more details regarding this.
- Service Health can store a maximum of 50 GB of monitoring data on the Crosswork Cluster. Crosswork Network Controller will raise an alarm when this storage reaches 70% of the 50 GB available storage capacity. In case you need additional storage, you can configure external storage in the cloud using an Amazon Web Services (AWS) cloud account. See [Configure Additional External Storage, on page 45](#).

Getting Started

Crosswork Service Health is available as part of the Cisco Crosswork Network Controller Advantage Package (see [Cisco Crosswork Network Controller Packages](#)).

To get started with Crosswork Service Health, follow the steps mentioned in the below table:



Note In order to set up and run Crosswork Service Health with Crosswork Network Controller, you only need to follow Steps 1 through 6 in the following table. Steps 7 to 9 are optional and explain advanced use cases of Crosswork Service Health.

| Workflow | Action |
|---|--|
| 1. Install Cisco Crosswork Network Controller Advantage package. | See the Cisco Crosswork Network Controller Installation Guide . |
| 2. Do the basic reachability checks from the Crosswork UI. | See Setup Workflow in the Cisco Crosswork Network Controller Administration Guide . |
| 3. Determine if you would like to configure additional external storage. Note You can configure external storage at any time. | If you anticipate monitoring health of many services, Cisco recommends configuring external storage after you install Crosswork Service Health and before you begin monitoring the services. See Workflow: Manage Stored Data, on page 6 . |
| 4. Create and provision the L2VPN/L3VPN services. | You can create and provision services using both the Crosswork Network Controller UI or using APIs: <ul style="list-style-type: none"> • Orchestrated Service Provisioning chapter in the Crosswork Network Controller Solution Workflow Guide. • Crosswork Network Controller API Documentation on Devnet |
| 5. Enable service health monitoring for the provisioned services. | Start monitoring VPN services. See Start Service Health monitoring, on page 11 . |

| Workflow | Action |
|--|---|
| 6. Establish your operational processes for responding to degraded services. | <p>Deep dive into the impacted services and sub-services health, and drill down to the root cause of the service degradation.</p> <p>See Workflow: Analyze the Cause of Service Degradation, on page 6.</p> |
| 7. (Optional) Use SR-PM to probe and monitor VPN services associated with TE policies. | <p>Use SR-PM to measure performance metrics of the underlay SR-TE policies to ensure that the VPN services are meeting the Service Level Agreements (SLA).</p> <p>See Workflow: View Performance Metrics of TE Policies using SR-PM, on page 7.</p> |
| 8. (Optional) Use Accedian Skylight to probe service health. | <p>Using Accedian Skylight probes can give additional insights into health of the service.</p> <p>See Workflow: Monitor Service Health using Accedian Skylight, on page 8.</p> <p>Note Accedian Skylight integration is available as a limited-availability feature in this release. Engage with your account team for more information.</p> |
| 9. (Optional) Customize and Import Heuristic Packages | <p>After you have used the default set of Heuristic packages that Service Health provides for monitoring, you may identify opportunities to customize them to better suit your needs.</p> <p>See Workflow: Customize Heuristic Packages, on page 9.</p> |

Service Health Workflows

In this section, we provide the details for each of the workflows identified in the [Getting Started, on page 4](#) section.

- [Workflow: Manage Stored Data, on page 6](#)
- [Workflow: Analyze the Cause of Service Degradation, on page 6](#)
- [Workflow: View Performance Metrics of TE Policies using SR-PM, on page 7](#)
- [Workflow: Monitor Service Health using Accedian Skylight, on page 8](#)
- [Workflow: Customize Heuristic Packages, on page 9](#)

Workflow: Manage Stored Data

Crosswork Service Health provides up to 50 GB of storage for monitoring data. If that limit is reached, the oldest monitoring data will be deleted first.

When the storage exceeds 70% capacity, Crosswork Network Controller generates an alarm prompting you to configure external storage in order to save older Service Health monitoring data. The actions detailed in the section describe how to monitor storage usage, reduce the amount of data being stored and how to add additional external storage.

Table 1: Workflow: Manage Stored Data

| Action | See.. |
|--|---|
| 1. Reduce the number of services being monitored by stopping the monitoring for few services. Review the monitoring data that is already stored on your system and delete any data that you no longer need to free up storage space. | Stop Service Health Monitoring, on page 16 |
| 2. Switch services that are using Advanced Monitoring to Basic Monitoring to monitor the services in lesser detail. | Edit Existing Monitoring Settings, on page 14 |
| 3. If you still need additional storage, configure additional external storage on AWS Cloud. | Configure Additional External Storage, on page 45 |

Workflow: Analyze the Cause of Service Degradation

This is an operational workflow and it is iterative. Deep dive into the impacted services and sub-services health, and drill down to the root cause of the service degradation in any of the following ways:

Table 2: Analyze the Cause of Service Degradation

| Action | See.. |
|--|--|
| 1. View Monitored Services and identify degraded services. | View Monitored Services, on page 29 |
| 2. Identify cause of the service degradation. | <ul style="list-style-type: none"> • Identify Root Causes Using Last 24Hr Metrics, on page 38 • Identify Active Symptoms and Root Causes of a Degraded Service, on page 32 • Identify Root Causes Using Assurance Graph, on page 35 |

| Action | See.. |
|---|---|
| <p>3. Confirm if the reported degradation is a valid issue. In case it is not a valid issue, you may need to adjust the monitoring level (from Basic Monitoring to Advanced Monitoring or vice versa) to ensure accurate reporting of service health.</p> <p>Alternatively, you can modify the system Heuristic package to create a custom Heuristic package to resolve the issue of false positive flagging of service health.</p> <p>If the reported issue is valid, proceed to the next step in this workflow.</p> | <p>Edit Existing Monitoring Settings, on page 14</p> <p>About Heuristic Packages, on page 49</p> |
| <p>3. Analyze if the service degradation is on account of an issue with device health.</p> | <ul style="list-style-type: none"> • View the Devices Participating in the Service, on page 41 • View Collection Jobs, on page 42 |

Workflow: View Performance Metrics of TE Policies using SR-PM

To measure the performance metrics of VPN services using either SR-MPLS or RSVP-TE Traffic Engineering policies, Service Health can leverage Segment Routing Performance Measurement (SR-PM). When this feature is enabled, Service Health gathers and processes additional metrics such as Delay, Delay Variance or Liveness to measure performance of the underlay SR-TE policy and determine Service Level Agreements (SLA) compliance.

The following workflow describes how you can enable SR-PM collection and view performance metrics of the underlay TE policies.

Table 3: Workflow to View Performance Metrics of TE policies using SR-PM

| Action | See.. |
|--|---|
| <p>1. Enable SR-PM metrics Collection in Crosswork and on the devices</p> | <p>Enable SR-PM Metrics Collection, on page 18</p> |
| <p>2. View the performance metrics of the policy</p> | <p>View SR-MPLS Policy Performance Metrics, on page 18</p> <p>View RSVP-TE Policy Performance Metrics, on page 19</p> |
| <p>3. Analyze the metrics and identify the cause of service degradation.</p> | <p>Identify Active Symptoms and Root Causes of a Degraded Service, on page 32</p> |

| Action | See.. |
|---|--|
| <p>4. Confirm if the reported degradation is a valid issue. In case it is not a valid issue, you may need to adjust the monitoring level (from Basic Monitoring to Advanced Monitoring or vice versa) to ensure accurate reporting of service health.</p> <p>Alternatively, you can create a custom Heuristic package by modifying the system Heuristic package for customized reporting of service health.</p> | <p>Edit Existing Monitoring Settings, on page 14</p> <p>About Heuristic Packages, on page 49</p> |

Workflow: Monitor Service Health using Accedian Skylight

Crosswork Network Controller can leverage external probing, provided by Accedian Skylight, to measure performance metrics of the L3VPN services. The metrics are compared with the contracted SLA (defined in the Heuristic package), and the results are made available on the UI for further analysis.



Note For the first time you add Accedian as a provider, follow step 1 and 2. Follow step 3 to 6 iteratively for operational purposes.

Table 4: Probe and Monitor Service Health using Accedian Skylight

| Action | See |
|---|--|
| 1. Install the Accedian Skylight Software. | <p>Refer to the Accedian Skylight documentation for information on installing Accedian Skylight and deploying it with Crosswork Network Controller.</p> <p>Note You need an account with Accedian Skylight to access the documentation. Sign up and create an account with the self sign-up tool.</p> |
| 2. Add Accedian Skylight as a provider in Crosswork Network Controller. | Add Accedian Skylight as a Provider, on page 21 |
| 3. Set up Probe sessions with Accedian Skylight for the L3VPN service. | Monitor Service Health using Accedian Skylight, on page 20 |
| 4. View the Accedian Skylight probe session Details in the Crosswork Network Controller UI. | View Accedian Skylight Probe Session Details, on page 22 |
| 5. Analyze the cause of the service degradation. | Identify Active Symptoms and Root Causes of a Degraded Service, on page 32 |

| Action | See |
|---|---|
| <p>6. Confirm if the reported degradation is a valid issue. In case it is not a valid issue, you may need to adjust the monitoring level (from Basic Monitoring to Advanced Monitoring or vice versa) to ensure accurate reporting of service health.</p> <p>Alternatively, you can modify the system Heuristic package to create a custom Heuristic package for a customized report of service health.</p> | <p>Edit Existing Monitoring Settings, on page 14</p> <p>Workflow: Customize Heuristic Packages, on page 9</p> |

Workflow: Customize Heuristic Packages

Crosswork Service Health uses Heuristic Packages as the core logic to monitor and report the health of services. Heuristic Packages define a list of rules, configuration profiles, supported sub-services and associated metrics for every service type. Heuristic Packages provided by the system are read-only and cannot be modified.

If you find that the Heuristic Packages provided by the system are insufficient in terms of monitoring metrics or monitoring thresholds, you have the option to export, modify and import the system package to create a customized Heuristic package that caters to your specific monitoring requirements.

Table 5: Customize Heuristic Packages

| Action | See.. |
|---|---|
| 1. Analyze your network services. Check the system Heuristic Packages for rules, sub-services, and metrics to ensure that the system packages do not have the required metrics, services or thresholds already. | See Basic and Advanced Monitoring Rules, on page 55 and Reference - Supported Subservices, on page 79 . |
| 2. Export and modify the Heuristic package to build a customized Heuristic package. | See About Heuristic Packages, on page 49 and Build a Custom Heuristic Package, on page 51 . |
| 3. Import the customized Heuristic package in Crosswork Network Controller. | Import Custom Heuristic Packages, on page 53 |
| 3. Apply the custom package to all the services that should be using it. | Start Service Health monitoring, on page 11 Edit Existing Monitoring Settings, on page 14 |
| 4. Verify that the custom package is providing the monitoring data that you need to meet your requirements. | View Monitored Services, on page 29 |



CHAPTER 3

Monitor Service Health

This section explains the following topics:

- [Start Service Health monitoring, on page 11](#)
- [Adjust Monitoring Settings, on page 14](#)
- [Enable SR PM Monitoring for TE Policies, on page 17](#)
- [Monitor Service Health using Accedian Skylight, on page 20](#)

Start Service Health monitoring

Before you begin

The following procedure assumes that you have already provisioned L2VPN/L3VPN services. To create and provision services, refer to the *Orchestrated Service Provisioning* chapter in the [Cisco Crosswork Network Controller Solution Workflow Guide](#).

To start health monitoring for a service:

-
- Step 1** From the main menu, choose **Services & Traffic Engineering > VPN Services**. The map opens on the left side of the page and the table opens on the right side.
- Step 2** In the Actions column, click for the service you want to start monitoring.
- Step 3** Click **Start Monitoring**.

VPN Services Refined By: All Endpoints ▾

Provisioning Health (Monitoring: 3 Services)

5 Success 0 Failed 0 In-Progress 2 Good 1 Degraded 0 Down

Total 5

Create ▾ ≡

| Health | Service ... | Type | Provisioni... | Las... [ⓘ] | Actions |
|--------|-------------|--------------|---------------|---------------------|------------------|
| | L2VPN_N... | L2vpn-Ser... | Success | 26-Jul-... | ... |
| | L3NM-PR... | L3vpn-Ser... | Success | 26-Jul-... | ... |
| | L3NM-PR... | L3vpn-Ser... | Success | 26-Jul-... | |
| | L3NM-PR... | L3vpn-Ser... | Success | 26-J | View Details |
| | L3NM-PR... | L3vpn-Ser... | Success | 26-J | Edit / Delete |
| | | | | | Start Monitoring |

Note The Health column color coding indicates the health of the service:

- Blue = Initiated
- Green = Good
- Orange = Degraded
- Red = Down
- Gray = Not Monitoring
- Yellow = Paused
- Red = Error

All services that are not being monitored currently have the **Health** column as gray.

Step 4 In the Monitor Service window that appears: .

- a) Select the **Monitoring Level** as **Basic Monitoring** or **Advanced Monitoring**.
- b) Click a Configuration Profile from the list of profiles that is displayed to select and apply it to monitor the service.

Monitor Service

Name: L3NM-PROBES-45-2-3-endpoint

Monitoring Level: ▼ ?

Silver_L3VPN_ConfigProfile custom

Gold_L3VPN_ConfigProfile custom

Basic Monitoring

Advanced Monitoring

Thresholds to use for Silver L3VPN services

Cpu Threshold Max: 80.5 %

Memfree Threshold Min: 1000000000 bytes

Cancel Start Monitoring

Step 5 Click **Start Monitoring**.

Note Once you have started monitoring the health of the service, in the Actions column, if you click ⋮ to view additional Service Health options, you will see: Stop Monitoring, Pause Monitoring, Edit Monitoring Settings, and Assurance Graph.

VPN Services Refined By: All Endpoints ▼

Provisioning: 5 ✓ Success, 0 ✗ Failed, 0 ⋮ In-Progress

Health (Monitoring: 3 Services): 2 ✓ Good, 1 ⚠ Degraded, 0 ⬇ Down

Total 5 ⚙️

Create ▼ ☰

| Health | Service ... | Type | Provisioni... | Las... ⓘ | Actions |
|----------------|-------------|--------------|------------------------|-----------------------|---|
| ✓ | L2VPN_N... | L2vpn-Ser... | ✓ Success | 26-Jul-... | ⋮ |
| ✓ | L3NM-PR... | L3vpn-Ser... | ✓ Success | 26-Jul-... | ⋮ |
| ⚠ | L3NM-PR... | L3vpn-Ser... | ✓ Success | | View Details |
| ⚠ | L3NM-PR... | L3vpn-Ser... | ✓ Success | | Edit / Delete |
| ⚠ | L3NM-PR... | L3vpn-Ser... | ✓ Success | | Stop Monitoring |
| | | | | | Pause Monitoring |
| | | | | | Edit Monitoring Settings |

After you start monitoring the service, the **Health** column of the service gets updated to reflected the health of the service.

What to do next

If Service Health reports the health as Degraded for the service, identify the root cause for service degradation and take measures to correct the issue. See [Analyze Degraded Services, on page 29](#) for more information.

Adjust Monitoring Settings


The following topics explain the various monitoring settings you can use to adjust the service health monitoring.

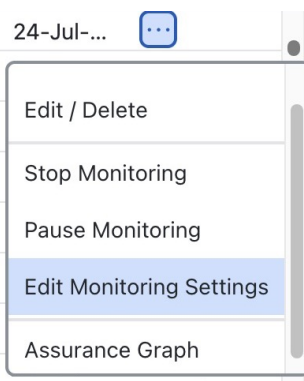
- [Edit Existing Monitoring Settings, on page 14](#)
- [Pause and Resume Service Health Monitoring, on page 15](#)
- [Stop Service Health Monitoring, on page 16](#)

Edit Existing Monitoring Settings

You can adjust the monitoring settings any time after the service health monitoring is enabled. You can update the Monitoring Level for the service from Basic Monitoring to Advanced Monitoring, or from Advanced Monitoring to Basic Monitoring. You can also update the Configuration Profile (from Gold profile to Silver profile or from Silver profile to Gold profile).

To edit the existing monitoring settings:

-
- Step 1** From the main menu, choose **Services & Traffic Engineering > VPN Services**. The map opens on the left side of the page and the table opens on the right side.
- Step 2** In the Actions column, click  for the service for which you want to edit the monitoring settings.
- Step 3** Choose **Edit Monitoring Settings** from the menu.



The Edit Monitoring Settings dialog box appears.

- Step 4** Choose the Monitoring Level or the Configuration Profile, as required.

Edit Monitoring Settings

Name L2-P2P-1381

Monitoring Level Basic Monitoring ?

Gold_L2VPN_ConfigProfile system | Gold_L2VPN_ConfigProfile system

Silver_L2VPN_ConfigProfile system | Thresholds to use for Gold L2VPN services

| | |
|-----------------------|------------------|
| Cpu Threshold Max | 70.5 % |
| Memfree Threshold Min | 2000000000 bytes |

Cancel Save

Note When switching between Advanced and Basic Monitoring, it may take over 15 minutes to view sub service health and active symptoms.

Step 5 Click **Save**.

A confirmation dialog box appears.

Step 6 Click **Start *monitoring-type* Monitoring**.

Crosswork Service Health starts monitoring the service health using the updated values.

What to do next


If Service Health reports the health as Degraded for the service, identify the root cause for service degradation and take measures to correct the issue. See [Analyze Degraded Services, on page 29](#) for more information.

Pause and Resume Service Health Monitoring

Using this option, you can temporarily pause monitoring the health of the services. This is useful in scenarios where a service is down due to a reported outage or scheduled maintenance and you do not want to continue to be notified about the degradation. If you pause and then resume monitoring a service, it resumes monitoring using the same Basic or Advanced monitoring rule and profile options that were used before the pause action. In addition, historical data and Events of Significance (EOS) are preserved in the history of the service. As no data is collected while monitoring is paused, the historical data will contain expected gaps when the monitoring is re-activated.

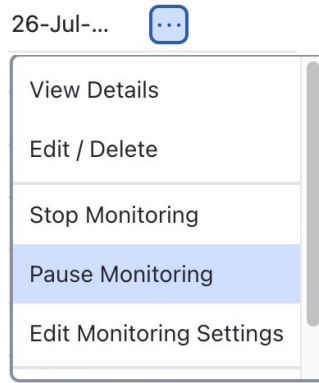
To pause and resume monitoring the health of the services, do the following:

Step 1 From the main menu, choose **Services & Traffic Engineering > VPN Services**. The map opens on the left side of the page and the table opens on the right side.


Step 2 In the Actions column, click  for the service that you want to pause the monitoring.

Step 3 Choose **Pause Monitoring** from the menu.

A confirmation dialog box appears. Click **Pause Monitoring**.



Note When monitoring is paused, you can still view the Assurance Graph which will show only the top level service with state of paused icon badge and with no child subservices underneath.

Step 4 In the Actions column, if you now click  for the service that you paused, you will see the **Resume Monitoring** option. Click this option to resume monitoring the service health.

A confirmation dialog box appears. Click **Resume Monitoring**.

Crosswork Service Health begins monitoring the health of the service using the same monitoring rule and profile options that were used before the Pause action.

Stop Service Health Monitoring

If you decide to stop monitoring a service, Service Health prompts you to confirm whether you want to retain the historical monitoring data or not. If you choose to retain the data, the historical data for the service will still be available if you start monitoring the service again. However, if you choose not to retain the historical data when you stop monitoring the service, all monitoring settings are deleted. The historical service data expires or purges if you choose to start monitoring the service later. In addition, the Assurance Graph for that stopped service will no longer be available. You may need to start monitoring the health of that service and begin service data collection anew.

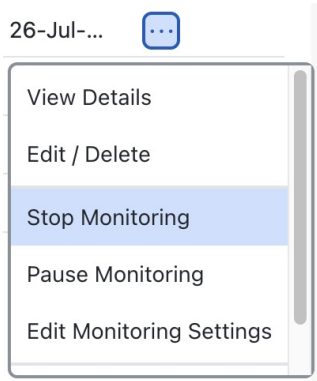
If you stop monitoring a service and do not select the **Retain historical Monitoring service for the data** check box, the monitoring settings are deleted.

To stop monitoring the health of a service, do the following:

Step 1 From the main menu, choose **Services & Traffic Engineering > VPN Services**. The map opens on the left side of the page and the table opens on the right side.

Step 2 In the **Actions** column, click  for the service you want to stop monitoring.


Step 3 Choose **Stop Monitoring** from the menu.



Step 4 The Stop Monitoring dialog box appears. To retain the historical service data for that service, select the **Retain historical Monitoring service for the data** check box.

Stop Monitoring

Name L3NM-PROBES-3403


 The health of the selected service will no longer be monitored and your monitoring settings will be deleted.
If you want to retain historical monitoring data select the checkbox below.
Are you sure you want to stop monitoring the health of this service?

Retain historical Monitoring service for the data

Cancel

Stop Monitoring

Step 5 Click **Stop Monitoring**.

Step 6 If you stopped monitoring a service and selected the **Retain historical Monitoring service for the data** check box, you can start monitoring that same service with historical data still available at a later time. From the **Actions** column of the service, click  and select **Start Monitoring**.

Note If External Storage is configured and you are monitoring a large number of services, you can ensure that the historical data of the stopped and restarted service is preserved for continued monitoring and inspection. For more information, see [Configure Additional External Storage, on page 45](#).

Enable SR PM Monitoring for TE Policies



To measure the performance metrics of VPN services associated with SR-MPLS or RSVP-TE Traffic Engineering policies, Service Health leverages the Segment Routing Performance Measurement (SR-PM) feature. This feature enables delay measurement and liveness detection on the underlay SR-TE policy to enforce Service Level Agreements in VPN services.

The SR-PM metric data is used for the policy subservice health computation and to determine the SLA for the service. You can view the KPI metrics, as well as the operational and administration status of the service,

on the policy tab in the **Service Details** page. The combination of probe metrics and administration and operational status helps determine the health of the subservice for the VPN instance. Additionally, this data is used to provide historical data for the policy and to indicate the end-to-end delay experienced by the traffic sent over an SR-TE or RSVP-TE policy. This information is used to determine whether the delay violated SLAs, which are defined in the Heuristic package.

Enable SR-PM Metrics Collection

To view the KPI metrics for the TE policies, you must first enable SR-PM metrics collection on the device for the TE policy and in Crosswork.

-
- Step 1** Configure SR-PM metrics collection in Crosswork.
- From the main menu choose **Administration > Settings > Performance Monitoring and Analytics > Performance Metrics**.
 - In the **Performance Metrics Settings** pane on the right, use the toggle button to enable **SR-PM collection for SR-Policies**.
 - Click **Save**.
- Step 2** Configure SR-PM collection on the device.
- Navigate to the Traffic Engineering Topology map. From the main menu choose **Services & Traffic Engineering > Traffic Engineering**.
 - Click the **SR-MPLS** or **RSVP-TE** tab as required.
 - Locate the policy you are interested in from the policy table. Click  and click **View Details**.
The policy details page opens.
 - In this page, click the Actions () button, and click **Edit/ Delete**.
Clicking this option will take you to the Provision (NSO) page and open the Edit Policy page.
 - Scroll down to the **performance measurement** section.
 - Toggle the **Enable performance-measurement** button.
 - Under **Profile-type**, click **delay** or **liveness** and click the toggle button in the profile to enable collection.

Note You can configure Delay or Liveness (not both together) manually on the device. See the device platform documentation for information. For example: [Segment Routing Configuration Guide for NCS 540 Series Routers](#).

If you enable the Delay, Delay Variance metrics, and disable the Liveness metric (or vice versa), the updated metric polling will take effect only from the next cadence. This is expected behavior.

View SR-MPLS Policy Performance Metrics

This procedure lists the steps for viewing KPI metrics for a SR-MPLS policy. KPI metrics contain Delay, Delay Variance (Jitter) or Liveness (Boolean value) along with traffic utilization.

Before you begin

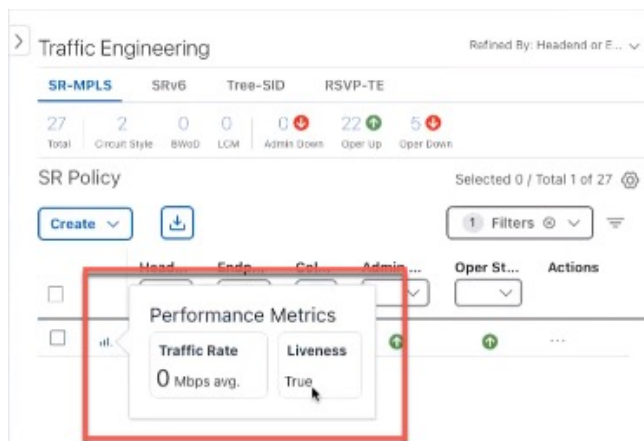
Ensure that devices and the policies have been added and device groups have been created.


Step 1 Navigate to the Traffic Engineering topology map. From the main menu, choose **Services & Traffic Engineering > Traffic Engineering**.

Step 2 Click the policy tab that you are interested in.

For example, to view policy performance metrics for SR-MPLS policies, click the SR-MPLS tab. Hover your mouse over the graph icon to view the KPI metrics in a carousel view.

Figure 1: SR-MPLS Policy Performance Metrics in the Traffic Engineering Table



Step 3 Alternatively, from the the **Actions** column, click  > **View Details** for one of the SR-MPLS policies. The **Service Details** page opens.

The KPI metrics for the policy are available in the **Performance Metrics** section.

View RSVP-TE Policy Performance Metrics

This procedure lists the steps for viewing KPI metrics for a RSVP-TE policy. KPI metrics include Delay and Delay Variance (Jitter) along with utilization.



Note The metric Liveness is not supported for RSVP-TE policies.

Before you begin

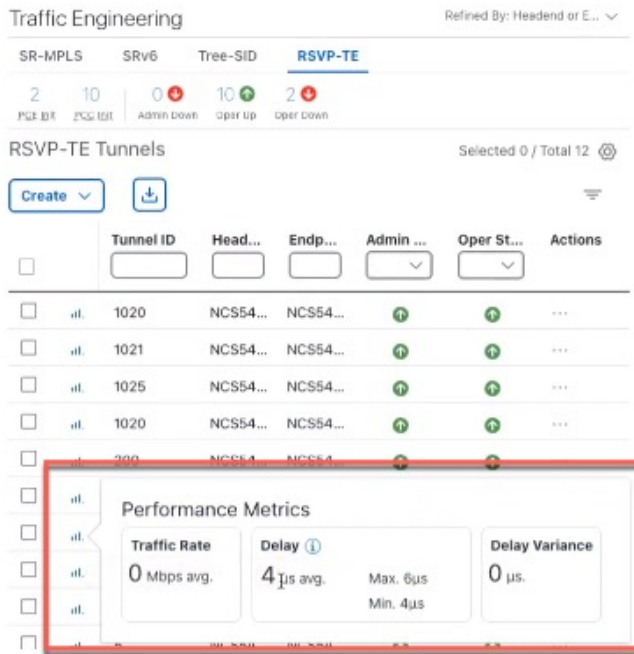
Ensure that devices and the policies have been added and device groups have been created.

Step 1 From the main menu, choose **Services & Traffic Engineering > Traffic Engineering**.

Step 2 From the **Traffic Engineering** window, select the **RSVP-TE** tab.

Hover your mouse over the graph icon to view the KPI metrics in a carousel view.

Figure 2: RSVP TE Tunnel Performance Metrics in the Traffic Engineering Table



Step 3 Alternatively, from the the **Actions** column, click > **View Details** for one of the RSVP-TE policies. The **Service Details** page opens.

The KPI metrics for the policy are available in the **Performance Metrics** section.

Monitor Service Health using Accedian Skylight

Crosswork Network Controller can leverage external probing, provided by Accedian Skylight, to measure metrics of the L3VPN services in the network. The metrics are compared with the contracted SLA (defined in the Heuristic package), and the results are made available on the UI for further analysis.

High-level Flow

1. When you provision a L3VPN service with probe intent and service monitoring is enabled, Accedian Skylight learns the probe intent and probe topology from provisioned service.

The following probe intents are supported:

- **Agent configurations:** ne-id, VLAN, IP, sub-interface.
- **Topology:** point-to-point, hub-spoke, full-mesh.

2. Probe sessions with Accedian Skylight are set up automatically to monitor the service by invoking the relevant RESTConf APIs. The list of RESTConf APIs that are invoked to provision probes sessions are

- endpoint, session, service, session activation. The maximum number of probe sessions per service are capped at 200 (for all connection types).

3. Service Health monitors a new sub-service *subservice.probe.session.health* and collects the following metrics for the service in the probe session:
 - Forward and Reverse Delay.
 - Forward and Reverse Variance.
 - Forward and Reverse Packet Loss.
4. The metrics collected during the probe sessions are analyzed and symptoms are raised accordingly, which are then displayed on the Crosswork Network Controller UI.

Add Accedian Skylight as a Provider

Before you begin

Ensure that you have taken care of the following prerequisites before onboarding Accedian Skylight as a provider:

1. Installed the Accedian Skylight software. Refer to the [Accedian Skylight documentation](#) for information on installing Accedian Skylight and deploying it with Crosswork Network Controller.



Note You need an account with Accedian Skylight to access the documentation. Sign up and create an account with the [self sign-up tool](#).

2. Have the following certificates from Accedian Skylight downloaded on your local system or on a folder that can be accessed by Crosswork Network Controller:
 - CA certificate
 - Client certificate
 - Client key

Step 1 Create a credential profile.

- a) Navigate to **Administration > Device Management > Credential Profiles** and click + to create a new profile.
- b) Enter a name, add the following credential protocols: **HTTPS** and **gNMI**. Add the username and password for both connections.
- c) Click **Save**.

Step 2 Create a certificate profile.

- a) Navigate to **Administration > Certificate Management** and click +.
- b) Enter a name and select the **Certificate Role as Accedian Provider Mutual Auth**
- c) Upload the certificates (ca_cert.pem, client_cert.pem, and client_key.key).
- d) (Optional) Enter the passphrase for the certificate chain.

e) Click **Save**.

Step 3 Add Accedian Skylight as a provider in Crosswork Network Controller.

a) Navigate to **Administration > Manage Provider Access**.

b) Click + and enter details in the fields as follows:

- **Provider Name:** Enter a name.
 - **Credential profile:** Select the credential profile that you created for Accedian Skylight.
 - **Family:** Select ACCEDIAN_PROXY.
 - **Certificate profile :** Select the Accedian Skylight certificate profile that you created in Step 2.
- Note** This field is displayed after you select the **Family** as ACCEDIAN_PROXY.
- **Connection types:** Supported protocols are automatically updated from the Accedian credential profile.
 - **IP addresses:** Enter the IP address or the Fully Qualified Domain Name (FQDN).
 - **Ports:** Enter 443 for HTTPS and a port value for GNMI.
 - **Encoding Type:** Select PROTO.

Note Only encoding of type **PROTO** is supported.

c) Click **Save**.

Step 4 Confirm reachability

a) From the main menu, choose **Administration > Manage Provider Access**.

b) Confirm that the Accedian Skylight provider shows a green Reachability status without any errors.

View Accedian Skylight Probe Session Details

To view the metrics from an Accedian Skylight probe session in the Crosswork Network Controller UI for a L3VPN service in the UI:

Before you begin

Ensure that you have completed the following steps:

1. Created and provisioned the required L3VPN services with the supported probe intent. For details, see the "Orchestrated Service Provisioning" chapter in the [Cisco Crosswork Network Controller Solution Workflow Guide](#)
2. Enabled service monitoring for the required services. See [Enable Service Health monitoring](#).

Step 1 Go to **Services & Traffic Engineering > VPN Services**. The map opens on the left side of the screen and the table opens on the right side of the screen.

Step 2 Click **Probe Sessions** tab in the **Service Details** page.

> Service Details
... | X

Name EP45-L3NM-IGP-405-Probe

Provisioning ✔ Success

Health ⚠ Degraded

Monitoring Status ✖ Error

Monitoring Settings Advanced | Gold_L3VPN_ConfigProfile custom ⓘ

Health
Transport
Configuration

↕ Path Query

Active Symptoms (8)
Probe Sessions (3)

Reactivate Probe
Filter 0 / Total 3 ⚙️ ☰

| Health | Probe ... | A Devi... | A Inter... | Z Device | Z Inter... | Actions |
|--|---|---|---|---|---|---|
| ▼ | | | | | | |
| ▬ ✔ | ✔ | CL4-PE... | Gigabit... | CL4-PE... | Gigabit... | ... |
| ▬ ✔ | ✔ | CL4-PE... | Gigabit... | CL4-PE... | Gigabit... | ... |
| ▬ ✔ | ✔ | CL4-PE... | Gigabit... | CL4-PE... | Gigabit... | ... |

Step 3 Click the graph icon next to a probe session for a detailed view of the performance metrics.

>

Probe Session Details

↻ | ✕

Details
Historical Data

Performance Metrics

Probe Delay
Forward

8.601 usec avg

Thresh.10000.000 usec

Probe Delay
Reverse

1.560 usec avg

Thresh.10000.000 usec

Probe Variance
Forward

10.361 usec avg

Thresh.2000.000 usec

< ● ● >

Summary

| | |
|------------------------|---|
| Service Name | EP45-L3NM-IGP-405-Probe |
| ProbeSession ID | 646c19e1-758a-529c-a870-6d3e39122355 |
| Subservice ID | ss-f6248e84-3205-480f-b251-5f1d111f8f4d |
| A Device | A CL4-PE-A |
| A Interface | GigabitEthernet0/0/0/0 |
| Z Device | Z CL4-PE-C |
| z Interface | GigabitEthernet0/0/0/0 |

The **Historical Data** tab provides probe metrics data from last 24-hours or from the service creation time (whichever is earlier) for all Probe metrics. See [View Historical Data from Accedian Skylight Probe Sessions, on page 26](#) for more information.

What to do next

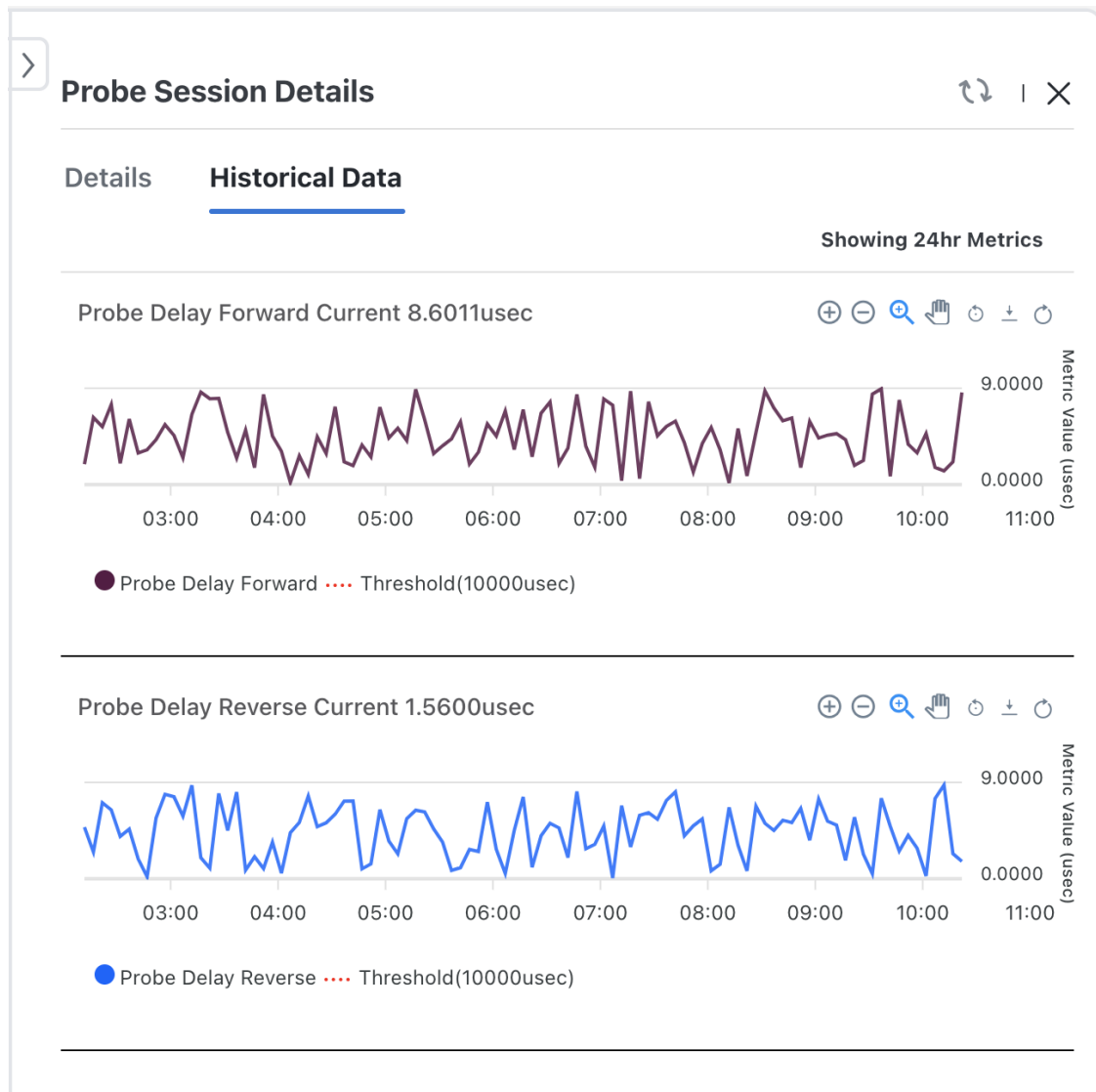
- Analyze the health of a degraded service. See [Analyze Degraded Services, on page 29](#) for more information.
- If a L3VPN service has probe provisioning errors, use the **Reactivate Probe** button to start the probe session again for the service. The **Probe Sessions** page gets updated automatically to reflect the updated metrics if the probe session was reactivated successfully.

View Historical Data from Accedian Skylight Probe Sessions

The **Historical Data** tab provides probe metrics data from last 24-hours or from the time the service monitoring was enabled (whichever is earlier) for all Probe metrics.



Note The database retains the data for up to 24 hours, and there may be some data retention for additional time before purging.



According to the current system behavior, when a monitoring session is deleted and re-added for a service, the **Historical Data** tab for that service will provide probe metrics data from the time monitoring was enabled or the last 24 hours (whichever is earlier) for all probe metrics. This is because the probe metric data is stored based on the devices and not based on the session or subservices.

Consider the scenario where monitoring for a service (consisting of up to 200 sessions) is enabled at 8:00 AM, and a monitoring session is removed at 9:00 AM, and the same session is re-added at 10:00 AM. At 9:00 AM the next day, you would expect to see historical data for this session from 10:00 AM the previous day. However, the actual behavior is that the historical data for the deleted and re-added session shows data from 9:00 AM the previous day.

Known Issues and Limitations with Accedian Skylight

The following is a list of known issues and limitations when Accedian Skylight is deployed for probing service health:

1. When monitoring is enabled for a service with probe intent but the Accedian Skylight provider is not added in Crosswork Network Controller, an error about the provider not being available is displayed for each of the probe metrics associated with the subservice.
2. You cannot delete the Accedian Skylight provider when a probe session is active.
3. The **Active Symptoms** tab displays the observed value of the metric at the time the symptom occurred, while the **Probe Sessions** tab is constantly updated with the live values of the metrics. Therefore, check the **Probe Sessions** tab for the real-time values of the performance metrics.
4. The Accedian Skylight provider is shown as reachable always in the Crosswork Network Controller Providers list page (**Administration > Manage Provider Access**) inspite of the following issues:
 - Connection issues between Accedian Skylight and Crosswork Network Controller.
 - Accedian Skylight provider credentials such as certificates, ports or IP addresses are wrong or invalid.

In such cases, services related to the Accedian Skylight probes have the health as degraded state with the symptom as '*Accedian provider does not exist in DLM*'. The symptoms are not cleared until you add the Accedian Skylight provider again, pause and resume the monitoring.



CHAPTER 4

Analyze Degraded Services

This section explains how Service Health monitoring helps to deep dive into the degraded services and subservices, and drill down to the root cause of the service degradation.

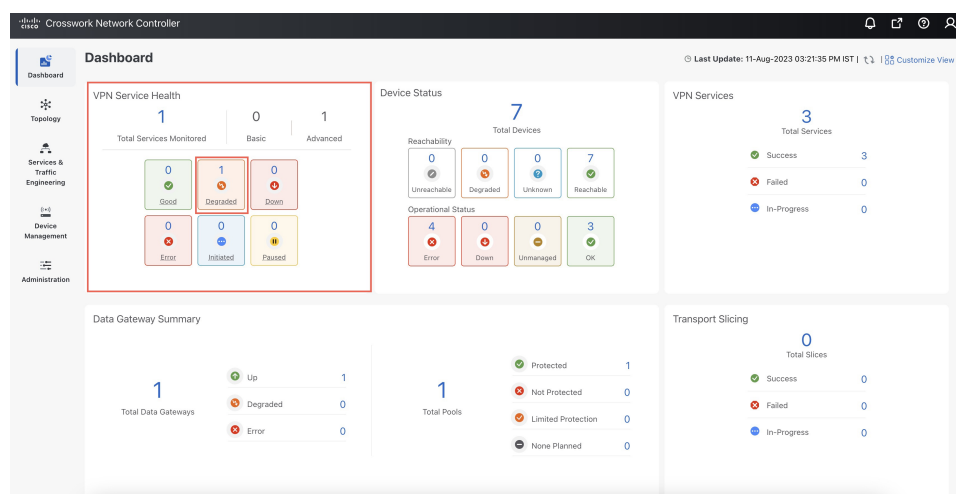
- [View Monitored Services, on page 29](#)
- [Identify Active Symptoms and Root Causes of a Degraded Service, on page 32](#)

View Monitored Services

You can view the monitored services in any of the following ways:

From the Crosswork Home Page

You will see the **VPN Service Health** dashlet on the Crosswork Home Page. This dashlet provides an overview of all the VPN services that are being monitored. From the dashlet you can click any of the status indicators to be taken to the **VPN Services** page with a filter set for the status you selected. To view the degraded services, click the **Degraded** box within the dashlet. This will take you to the VPN Services page, where only the degraded VPN services are displayed.









From the VPN Services Page


From the main menu, choose **Services & Traffic Engineering > VPN Services**. All the VPN services are listed on this page. The degraded services show an orange icon in the **Health** column.

You can filter the services by their health (Down, Degraded, Good, Paused, Initiated, Error, Unmonitored). You can also click the Degraded tab in the Health tab in this page to filter and view only the Degraded services.








VPN Services Refined By: All Endpoints ▾

Provisioning Health (Monitoring: 1 Services)

3  Success 0  Failed 0  In-Progress 0  Good 1  Degraded 0  Down


Total 3 

Create ▾ ☰

| Health | Service ... | Type | Provisioni... | Las...  | Actions |
|---|----------------------|--------------|---|--|---------|
| ▾ | <input type="text"/> | ▾ | ▾ | | |
|  | L2VPN_N... | L2vpn-Ser... |  Success | 02-Aug... | ... |
|  | L3NM-PR... | L3vpn-Ser... |  Success | 09-Aug... | ... |
|  | L3NM-PR... | L3vpn-Ser... |  Success | 06-Aug... | ... |

To clear the filter, click **X** next to the designated filter appearing in the space at the top of the column. To remove all the filters and to show all the VPN services, click the **X** icon in the Filters field above the table.



Note If a service is not yet being monitored, a gray icon is displayed in the Health column. To enable monitoring for such a service, click  and select **Start Monitoring**. For more information, see [Start Service Health monitoring, on page 11](#).

View Monitoring Status of a Service

You can view the **Monitoring Status** of a service from its **Service Details** page.

From the main menu, choose **Services & Traffic Engineering > VPN Services**. Locate the service that you are interested in and under the **Actions** column, click **View Details**. This page displays the **Monitoring Status** and the **Health** status of the service.

Name L3VPN_U_NM-SRTE-ODN-1061

Provisioning ✔ Success

Health ⚠ Degraded

Monitoring Status ✖ Error

Monitoring Settings Advanced | Gold_L3VPN_ConfigProfile custom ⓘ

Health Transport Configuration Path Query

Active Symptoms (4) Probe Sessions (0)

4 All 2 Symptoms 2 Monitoring Errors Total 4 ⚙️ ☰

| Root Cause ⓘ | Subservice | Type | Prior... | La: |
|----------------------|-----------------------|-------------------|----------|-----|
| Unable to get fee... | subservice.interfa... | Monitoring Errors | 2 | 31- |
| Unable to get fee... | subservice.interfa... | Monitoring Errors | 2 | 31- |
| eBGP Session to ... | subservice.ebgp.n... | Symptoms | 255 | 31- |
| eBGP Session to ... | subservice.ebgp.n... | Symptoms | 255 | 31- |

Monitoring status for a service can be either **Healthy** or **Error**.

- **Healthy:** This means the end-to-end flow of monitoring the service is working as expected and Service Health is able to evaluate the health of the service successfully.
- **Error:** This means Service Health is unable to monitor the current health of the service due to component failures, operational errors or device errors, and the health status that is displayed is the last known health of the service. You can filter monitoring errors using the mini dashboard or the filters.



Note Monitoring errors reported on account of device health do not affect the overall health of the service.

In the Historical Graph, Events of Significance (EoS) are displayed for monitoring errors as well. If the service is healthy but there are monitoring errors, a green warning icon is displayed. However, if the service is degraded and there are monitoring errors, then an orange warning icon is displayed. Clicking these icons provides you with the details in the symptoms table with type as **Monitoring Errors**.



Note The Historical Graph displays monitoring errors only when the Monitoring Errors setting is enabled via API. There is no option to enable this setting from the UI in this release. Once this setting is enabled, the system starts to log these monitoring errors as Events of Significance and display them in the historical graph. Refer to the [API documentation on Cisco Devnet](#) for more information.


Identify Active Symptoms and Root Causes of a Degraded Service

By analysing the root cause of reported active symptoms and impacted services, you can determine what issues must be addressed first to maintain a healthy setup and what requires further inspection and troubleshooting.



Note L3VPN service monitoring is supported on Cisco IOS XR devices and not on Cisco IOS XE devices. For an L3VPN service being monitored, if a provider and devices are deleted, and then added again, the monitoring status will remain in the degraded state with a monitoring status as Monitoring error. Stop and restart the service monitoring to recover from this error.

To view the active symptoms and root causes for a service degradation:

-
- Step 1** From the main menu, choose **Services & Traffic Engineering > VPN Services**. The service assurance dependency graph opens on the left side of the page and the table opens on the right side.
- Step 2** In the Actions column, click  and click **View Details**. The Service Details panel appears on the right side.
- Step 3** Select the Health tab and click the **Active Symptoms** tab. The Active Symptoms table displays **Active Symptoms** and **Monitoring Errors** by default. To filter the table to show only the Active Symptoms, either click the **Symptoms** tab in the mini dashboard above the table or select **Symptoms** from the filter box under the **Type**. The table now shows a filtered list containing only the Active Symptoms.
- Review the Active Symptoms for the degraded service (including the Root Cause, Subservice, Type, Priority, and Last Updated details).

Service Details

Name P2P-SR-C-101
Provisioning ✔ Success
Health ⚠ Degraded
Monitoring Status ✘ Error
Monitoring Settings Advanced | Gold_L2VPN_ConfigProfile system ⓘ

[Health](#) | [Transport](#) | [Configuration](#) 🔗 Path Query

Active Symptoms (16) | **Probe Sessions (0)**

Total 16 ⚙️

16 All | 8 Symptoms | 8 Monitoring Errors

| Root Cause ? | Subservice | Type | Prior... ↑ | Li |
|-------------------------------------|-----------------------|----------|------------|----|
| PCEP Session He... | subservice.pcep.s... | Symptoms | 10 | |
| PCEP Session He... | subservice.pcep.s... | Symptoms | 10 | |
| VPWS State degr... | subservice.vpws.c... | Symptoms | 15 | |
| VPWS State degr... | subservice.vpws.c... | Symptoms | 15 | |
| Fallback LSP pat... | subservice.p2p.fal... | Symptoms | 255 | |

Step 4

Click on a Root Cause and view both the **Symptom Details** and the **Failed Subexpressions & Metrics** information. You can expand or collapse all of the symptoms listed in the tree, as required. In addition, use the **Show Only Failed** toggle to focus only on the failed expression values.

Service Details ... X

Name P2P-SR-C-101

Provisioning ✓ Success

Health ⚠ Degraded

Monitoring Status ✗ Error

Monitoring Settings Advanced | Gold_L2VPN_ConfigProfile system ⓘ

Health | Transport | Configuration ↕ Path Query

Symptom Details ^

Name PCEP Session Health degraded. Device: CL2-PE-C, PCC-Peer: 192.168.15.42

Sub Service subservice.pcep.session.health system

Last Updated 28-Jul-2023 11:29:20 PM IST

Failed Subexpressions & Metrics ^

Show Only Failed Expand All | Collapse All

| Name | Expre |
|---------------------------------------|-------|
| explabel | pcc_p |
| ⚠ pcc_peer_state == 'up' | false |

Step 5 Click the **Transport** and **Configuration** tabs and review the details provided.

Step 6 Click **X** in the top-right corner to return to the VPN Services list.

Related Information

- To monitor the VPN services using Assurance Graph capabilities and inspect any services or related nodes that are degraded, see [Identify Root Causes Using Assurance Graph, on page 35](#).
- To identify the issues with the degraded services within a specific time range, use the Last 24Hr Metrics. For details, see [Identify Root Causes Using Last 24Hr Metrics, on page 38](#).
- To identify a service health issue by examining the collection jobs, see [View Collection Jobs, on page 42](#).

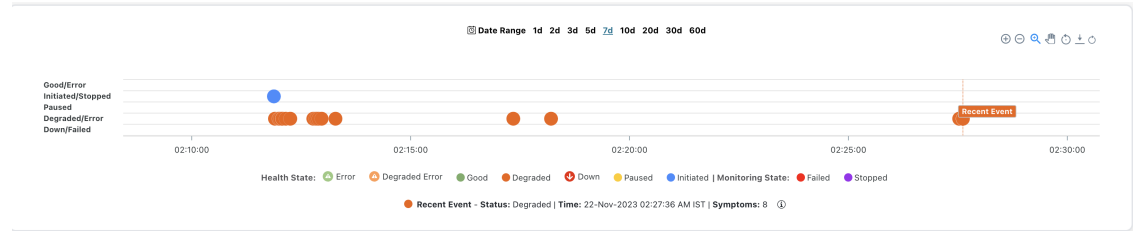
About Assurance Graph

In Crosswork Service Health, a service instance is decomposed into subservices, which are then assured independently. Assurance Graph represents the service instances, and their dependent subservices in a graphical form. In this logical dependency tree, the topmost node represents the service instance that is being monitored and its child nodes represent the components identified as its subservices. Each subservice can have

dependencies as well. Assurance Graph helps to locate the problem area, and provides an indication of the possible symptoms and impacting metrics, which in turn helps in troubleshooting in case of degradation.

Crosswork Service Health updates the Assurance Graph automatically when the service instance is modified.

To view a service in the Assurance Graph, from the **Actions** column for the service, select **Assurance Graph**. Toggle the **Show History** button to view the historical data chart. Each dot on the history chart represents one Event Of Significance (EOS) for a service.



For each EOS, you can view the Assurance Graph and symptoms with 24 hours metrics collected based on the time of the EOS. For example, for a service for which monitoring was stopped, a dot appears indicating that the monitoring was stopped. Using your mouse, click and drag over a selected range over the EOS to zoom in on the time range. Hover your mouse over the service to view details about the event and any associated symptoms.

Identify Root Causes Using Assurance Graph

You can monitor the VPN services using Assurance Graph capabilities and inspect any services or related nodes that are degraded.

Before you begin

- Ensure that service health monitoring is enabled for the service you want to inspect. For details, see [Start Service Health monitoring, on page 11](#).
- Before using Service Health's Assurance Graph, ensure that topology map nodes have been fully configured and created with a profile associated to the service. If not, Subservice Details metrics will show that no value has yet to be reported.




Note L3VPN service monitoring is supported on Cisco IOS XR devices and not on Cisco IOS XE devices. For an L3VPN service being monitored, if a provider and devices are deleted, and then added again, the monitoring status will remain in the degraded state with a monitoring status as Monitoring error. Stop and restart the service monitoring to recover from this error.

To identify the root causes using Assurance Graph, do the following:

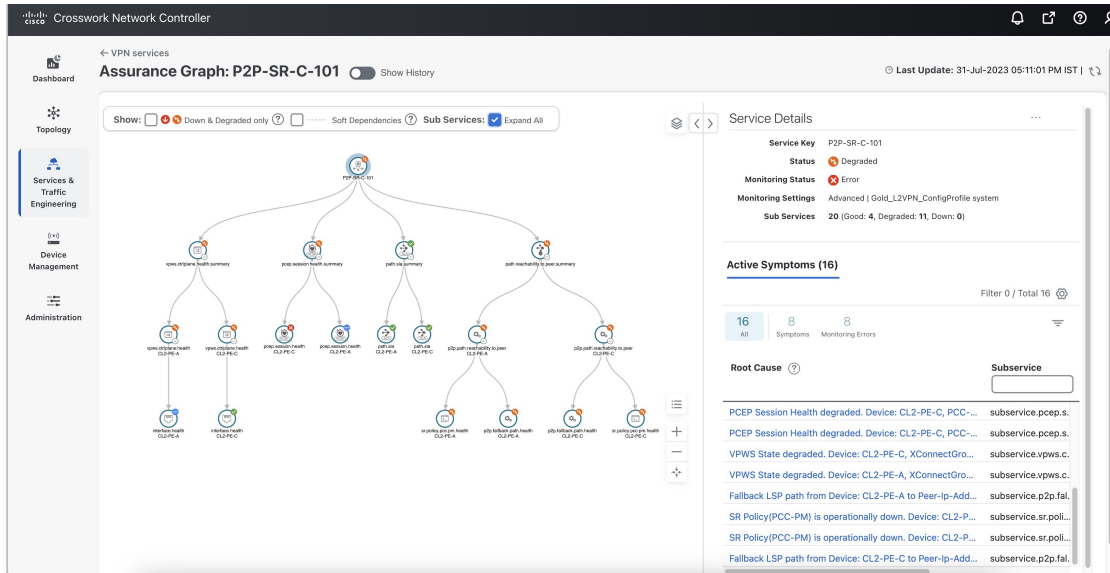
Step 1

From the main menu, choose **Services & Traffic Engineering > VPN Services**. The service assurance dependency graph view opens on the left side of the page and the table opens on the right side.

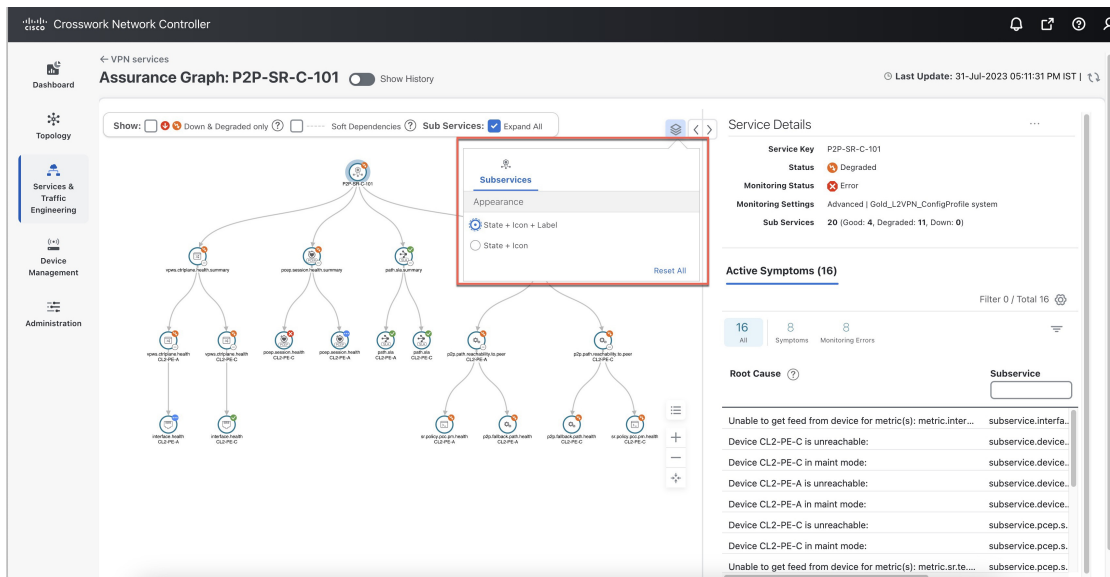
Identify Root Causes Using Assurance Graph

Step 2 In the Actions column, click  for the required degraded service and click **Assurance Graph**. The service assurance dependency graph view of services and subservices appear with the Service Details panel showing Service Key, Status, Monitoring Status, Monitoring Settings, Sub Services, and Active Symptoms details.

This may take up to 5-10 minutes to update after a service has been enabled for monitoring.



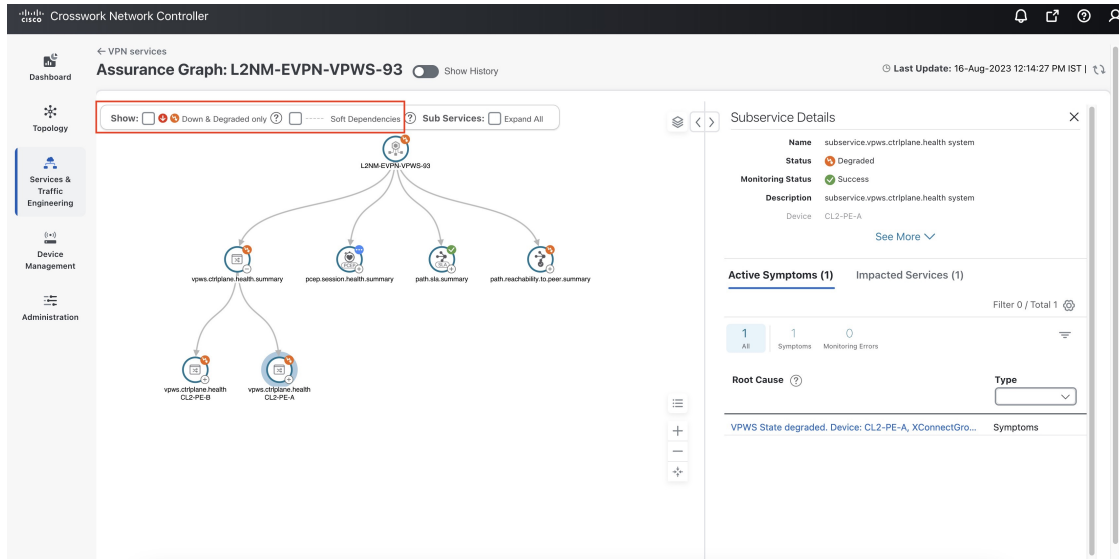
At the top-right of the service assurance dependency graph, select the stack icon to select the appearance option for the Subservices: **State + Icon + Label** or **State + Icon**.



Step 3 By default, the Assurance Graph displays a concise view with only the service and the top level subservices (aggregator nodes). Click the + icon in the nodes to expand the graph and to view the dependent details. To expand all the nodes at once, click the **Subservices: Expand All** check box at the top.

Step 4 Select a degraded subservice in the Assurance Graph. The Subservice Details panel appears with subservice metrics, as well as subservice specific Active Symptoms and Impacted Services details.

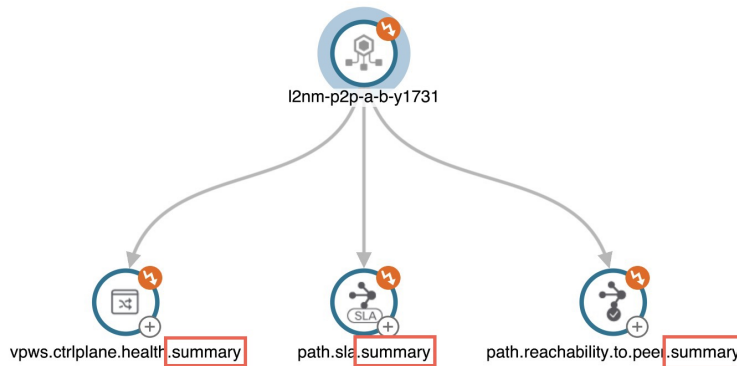
- **Active Symptoms:** Provides symptom details for nodes actively being monitored.
- **Impacted Services:** Provides information for services that are impacted by issues based on historical monitoring of health status.



Note At the top left of the service assurance dependency graph, check the **Down & Degraded only** or **Soft Dependencies** check boxes to further isolate the subservices. Soft Dependencies implies that a child subservice's health has a weak correlation to its parent's health. As a result, the degraded health of the child will not result in the parent's health degradation.

Note In some cases, the Summary node feature is available and summarizes the aggregated health status of child subservices and reports a consolidated health status to a service node. The Summary node feature is available in both L2VPN multipoint Basic and Advanced monitoring models.

- Basic monitoring subservices:
 - Device—Summarizes the health status of all underlying Devices participating in the given L2VPN service.
 - Bridge Domain—Summarizes the L2VPN service's Bridge Domain health status across all participating devices.
- Advanced monitoring subservices (in addition to what is also available with Basic monitoring):
 - EVPN—Summarizes the health status of all underlying subservices—BGP Neighbor Health and MacLearning Health across all participating PE endpoints and provides a consolidated overall EVPN health summary status.
 - Transport—Summarizes the health status of all underlying subservices—SR-ODN (dynamic), SR Policy (statically configured), and RSVP TE Tunnel, across all participating PE endpoints and provides a consolidated overall Transport health summary status.
 - SR-PCEP—Summarizes the health status of all the underlying subservices that are monitoring the PCEP sessions. Each underlying subservice monitors the PCEP session health on a particular device participating in the given VPN service.



Step 5 Inspect the Active Symptoms and Impacted Services information, and the root causes associated with the degraded service to determine what issues may need to be addressed to maintain a healthy setup.

Related Information

- To view the Active Symptoms and Root Causes, see [Identify Active Symptoms and Root Causes of a Degraded Service, on page 32](#).
- To identify the issues with the degraded services within a specific time range, use the Last 24Hr Metrics. For details, see [Identify Root Causes Using Last 24Hr Metrics, on page 38](#).
- To identify a service health issue by examining the collection jobs, see [View Collection Jobs, on page 42](#).

Identify Root Causes Using Last 24Hr Metrics


You can utilize the Last 24Hr Metrics to identify the issues with the degraded services within a specific time range. By isolating the issues within a specific time range, you can drill down on the details that may have caused the degraded (or down) service that can lead to troubleshooting the service or the node to address detailed symptoms.

Before you begin

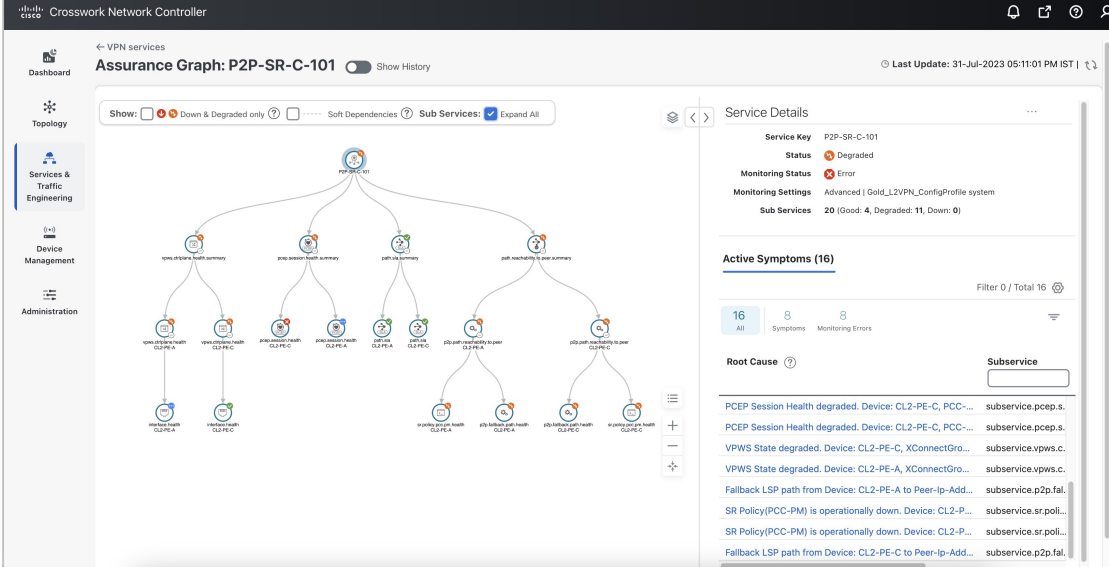
- Ensure that service health monitoring is enabled for the service you want to analyze. For details, see [Start Service Health monitoring, on page 11](#).
- Before using Service Health's Assurance Graph feature, ensure that topology map nodes have been fully configured and created with a profile associated with the service. If not, Subservice Details metrics will show that no value has yet to be reported.



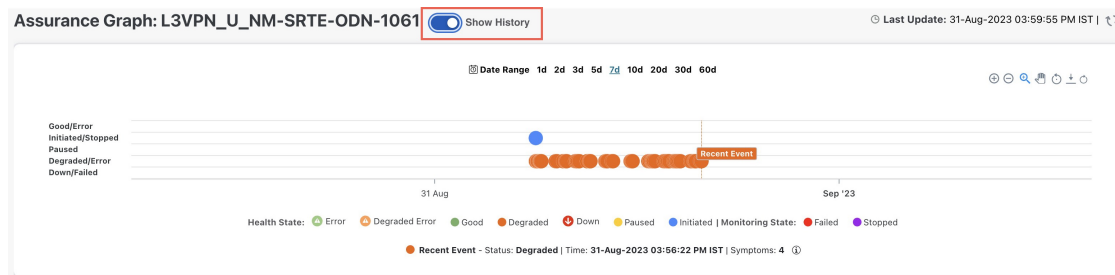
Note L3VPN service monitoring is supported on Cisco IOS XR devices and not on Cisco IOS XE devices. For an L3VPN service being monitored, if a provider and devices are deleted, and then added again, the monitoring status will remain in the degraded state with a monitoring status as Monitoring error. Stop and restart the service monitoring to recover from this error.

- Step 1** From the main menu, choose **Services & Traffic Engineering > VPN Services**. The service assurance dependency graph opens on the left side of the page and the table opens on the right side.
- Step 2** In the Actions column, click  for the degraded service and click **Assurance Graph**. The service assurance dependency graph of services and subservices appear with the Service Details panel showing Service Key, Status, Monitoring Status, Monitoring Settings, Sub Services, and Active Symptoms details.

Note This may take up to 5-10 minutes to update after a service has been enabled for monitoring.



- Step 3** At the top of the page, click the **Show History** mode toggle. The historical Date Range graph appears. This graph shows different ranges of historical health service monitoring details from one day (1d) up to sixty days (60d). When you hover over an event on the Date Range graph, a tool tip with information about that event appears (such as date and time of the event, and number of symptoms).



- Step 4** Review the Root Cause information by clicking a particular event in the graph. The Service Details panel reloads, showing the active symptoms and the root causes associated with the event. Columns can be resized using your mouse or you can select the gear icon to deselect or select columns you want to appear.

Note Once you enable **Show History** mode, Root Cause information in the Active Symptoms table will start to show the blue Last 24Hr Metrics icon. Data from the device will be initially delayed, however, and may take some time before **Last 24Hr Metrics** begins to populate with data. Until then, the value of zero is reported.

Service Details

Service Key L3VPN_U_NM-SRTE-ODN-1061

Status ⚠ Degraded

Monitoring Status ✖ Error

Monitoring Settings Advanced | Gold_L3VPN_ConfigProfile custom

Sub Services 27 (Good: 19, Degraded: 5, Down: 0)

Symptoms (4)

4 All | 2 Symptoms | 2 Monitoring Errors | Total 4

Root Cause ⓘ | **Subservice**

| Root Cause | Subservice |
|---|-----------------------|
| Unable to get feed from device for metric(s): metric.inter... | subservice.interfa... |
| Unable to get feed from device for metric(s): metric.inter... | subservice.interfa... |
| eBGP Session to neighbor 10.10.10.238 is not up for D... | Last 24Hr Metrics ... |
| eBGP Session to neighbor 10.10.10.238 is not up for Device: CL2-PE-A, Vrf: L3VPN_U_NM-SRTE-ODN-1061 | |

Step 5

To further isolate the possible issues and to utilize the **Last 24Hr Metrics**, perform the following steps:

- In the Date Range graph, use your mouse to select the range of historical health service monitoring details from one day (1d) up to sixty days (60d).

Note At the top-right of the Date Range graph, select the appropriate icons to either zoom in or out, horizontally scroll through the date ranges, or refresh the graph to go back to the most recent event, for example. You can also use your mouse to draw a rectangle over events to further zoom in on the degraded devices. Events that are consecutive may appear as a line of white space.

- Click on a degraded event in the graph. The Service Details panel reloads, showing any active symptoms and the root causes to be inspected. Expand the table and information as necessary for further details.


Step 6

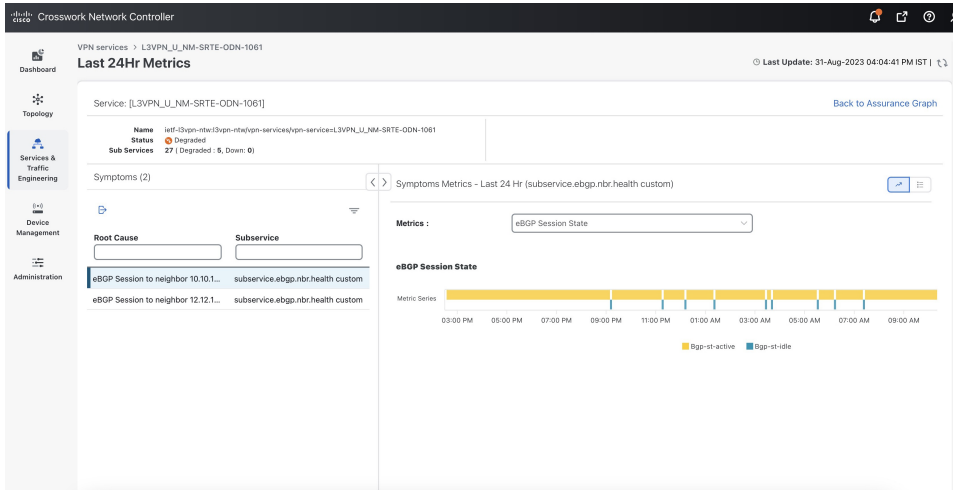
Check the **Down & Degraded Only** check box at the top-left corner of the Assurance graph to show only the Subservices which are degraded, along with other dependent but healthy subservices. Inspect the Service Details panel showing the active symptoms and their root cause. Uncheck the **Down & Degraded Only** check box and check the **Soft Dependencies** check box in the top-left corner of the Assurance graph. Soft Dependencies implies that a child subservice's health has a weak correlation to its parent's health. As a result, the degraded health of the child will not result in the parent's health degradation.

Use the + or – symbols in the bottom-right corner of the Assurance graph to zoom in or out on sub-services mapped. Select the ? to view the Link Color Legend that explains all of the icons, symbols, badges, and colors and their definitions.

Step 7

Select the degraded subservice in the Assurance graph to show the subservice details.

- Step 8** Click the **Symptoms** tab to show any root causes for the service health details that are displayed and then click the **Impacted Services** tab to view the impacted services.
- Step 9** Click **X** in the top-right corner to return to the VPN Services list and in the Actions column, click  for the degraded service in the list and click **Assurance Graph** to show the Service Details panel.
- Step 10** Again, select the **Show History** toggle in the top-right corner of the Service Details panel before selecting the blue metrics icon in one of the Root Cause rows. The Symptoms Metrics – Last 24 Hr bar chart appears. This chart provides details of the metric patterns for different sessions states (such as active, idle, failed) for individual root cause symptoms with Status, Session, Start Time, and Duration information to assist in troubleshooting prevailing issues. Use your mouse to hover over the chart to view the different details.



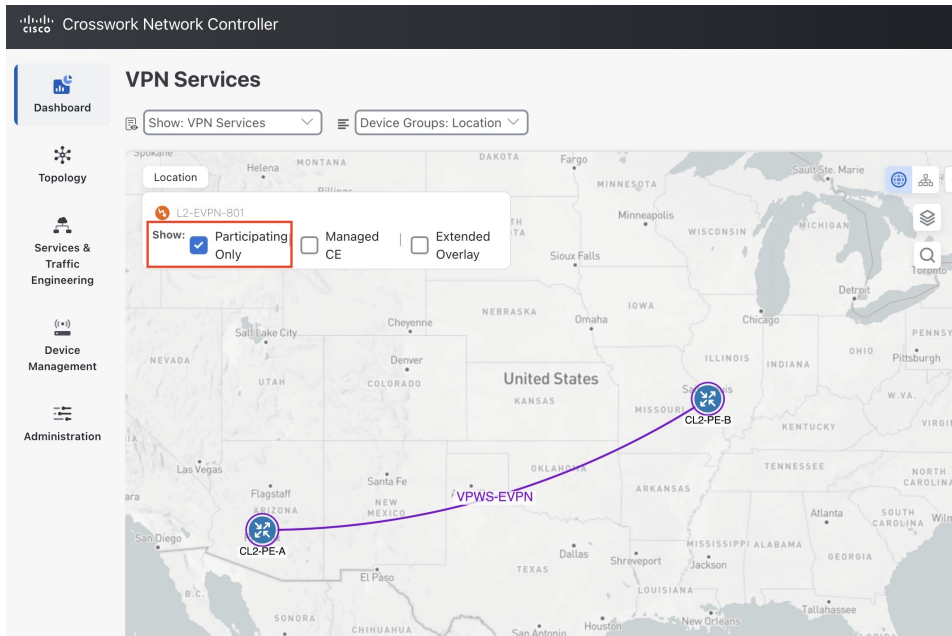
Related Information

- To view the Active Symptoms and Root Causes, see [Identify Active Symptoms and Root Causes of a Degraded Service, on page 32](#).
- To monitor the VPN services using Assurance Graph capabilities and inspect any services or related nodes that are degraded, see [Identify Root Causes Using Assurance Graph, on page 35](#).
- To identify a service health issue by examining the collection jobs, see [View Collection Jobs, on page 42](#).

View the Devices Participating in the Service

When a device or interface related subservice degrades, the corresponding devices display an orange icon in the topology view. To view the devices participating in the services, do the following:

- Step 1** From the main menu, choose **Services & Traffic Engineering > VPN Services**. The service assurance dependency graph view opens on the left side of the page and the table opens on the right side.
- Step 2** Click on the name of a service that shows as being degraded. The service assurance dependency graph view is updated, isolating the corresponding devices participating in that service.
- Step 3** At the top-left of the service assurance dependency graph view, select the **Show Participating Only** check box so that the service assurance dependency graph only shows the devices participating in the service.



Step 4 Hover your mouse over the device icons and review the popup information relating to its Reachability State, Host Name, Node IP, and Type.

The devices that are healthy may show an orange badge to indicate that there are device or interface related subservices underneath that are not healthy. This ensures that unhealthy subservices are easily visible and can be identified from the topological view even if the device itself is healthy. After examining the Service Details for a device, for example, a condition, such as the CPU is low on a subservice node, helps to take the necessary steps to address the unhealthy subservice.

View Collection Jobs

Crosswork Service Health provides the capability to view Parameterized Jobs (template-based collection jobs) that supports a greater number of jobs, adding the ability to view CLI collection jobs. This is useful when troubleshooting collection job issues by examining details of individual devices. Devices are identified by their Context ID (protocol) to determine if the jobs are gNMI, SNMP, or CLI-based jobs.


The **Parameterized Jobs** tab on the Collection Jobs page lists all active jobs created by the Cisco Crosswork Service Health application. If Crosswork Service Health is not deployed, this page will have no data.

Step 1 From the main menu, choose **Administration > Collection Jobs**.

The Collection Jobs page appears.

Step 2 Click the **Parameterized Jobs** tab.

Step 3 Review the Parameterized Jobs list to identify the devices that may have service health degradation issues. By reviewing Parameterized Jobs, you can identify and focus on gNMI, SNMP, and CLI-based jobs by their Context ID (protocol) for further troubleshooting purposes.

- Step 4** In the Job Details panel, select the collection job you want to export and click  to download the status of collection jobs for further examination. The information provided is collected in a .csv file when the export is initiated.
- Note** When exporting the collection status, you must fill in the information each time an export is executed. In addition, make sure to review the **Steps to Decrypt Exported File** content available on the Export Collection Status dialog box to ensure you can access and view the exported information.
- Step 5** Click **Export**.
- Step 6** To check the status of the exported collection job data, click **View Export Status** at the top right of the Job Details panel. The Export Status Jobs panel appears providing the status of the export request.
- Step 7** Review the exported .csv file for collection job details and the possible cause of the degraded device.
-



CHAPTER 5

Configure Additional Storage

This section explains the following topic:

- [Configure Additional External Storage, on page 45](#)

Configure Additional External Storage

Crosswork Service Health provides internal storage of monitored data up to a maximum limit of 50 GB. This data is stored on your system. If you exceed the limit of the internal storage, historical data will be deleted.

If you anticipate monitoring health of many services, Cisco recommends configuring external storage after you install Crosswork Service Health and before you begin monitoring the services. When the storage reaches 70% capacity, Crosswork Network Controller generates an alarm prompting you to configure external storage in order to save older Service Health monitoring data. If you do not configure external storage, the oldest files are deleted when 80% of 50 GB storage capacity is reached.

By leveraging external storage, all existing internal storage data will be automatically moved to the external cloud storage and your internal storage will act locally as cache storage. Configuring external storage for Crosswork Service Health ensures that you do not lose historical data for services that continue to monitor a service's health. Also, it ensures the service health data is retained for any service you choose to stop monitoring when you select the option to retain historical monitoring service for the data. For more information on how to retain the historical monitoring service data when stopped, see [Stop Service Health Monitoring, on page 16](#).

You can use an Amazon Web Services (AWS) cloud account to configure external storage in the cloud. Only AWS S3, which is object storage, is supported. The data is stored by Service Health in the tar.gz archive file format. This data includes the VPN service status at the time of storage and historical data about the service. Each tar.gz file represents an EoS (Event of Significance). Service Health uses this data to display it visually in the Crosswork Network Controller UI when you click on an EoS.

After you configure AWS storage, only 80% of the 50 GB space or 100,000 files are stored locally in Crosswork Network Controller. The oldest files are automatically moved to AWS.

Before you begin

You must have an AWS cloud account set up so to configure the external storage.

Step 1 From the main menu, choose **Administration > Settings** and click the **Storage Settings** tab.

Crosswork Network Controller

Settings

System Settings User Settings **Storage Settings**

Overview Configuration Diagnostics Jobs

Internal Storage

Used 968.76 MB Free 52.72 GB 53.69 GB

External Storage

There is no data to view. Configure to view External info.

[Configure](#)

Step 2 With the Overview tab selected, click **Configure** under the **External Storage** section. The Configuration page appears with the Data Storage Type and S3 Provider fields pre-populated with AWS.

System Settings User Settings **Storage Settings**

Overview **Configuration** Diagnostics Jobs

Data Storage Type *

S3 Provider *

Access Key *

Secret Key *

End Point * ⓘ

Region * ⓘ

Bucket *

Advance Settings

Storage Class * ⓘ

Expiry Period / days

Http Proxy ⓘ

Transfer Acceleration Enable Disable

ⓘ Files in local cache will be bulk copied over to external storage, this will allow incremental uploads for the new files improving application performance

Copy Local Data

[Test & Save](#)

Step 3 Provide your AWS authentication information for all of the required fields (such as Access Key, Secret Key, End Point, and so on).

Step 4 Check the **Copy Local Data** check box if you want all files, previously stored in the local cache, to be bulk copied to the external storage. This action will allow for incremental upload of the new files.

Note This option is a one-time action when moving from only maintaining local storage and moving to external storage. This action also helps to improve the application performance.

Note The Expiry Period refers to the number of days that historical data files will be stored before being deleted. For example, if the Expiry Period is set to 1, the files will be deleted two days later, at midnight of the operational time zone of the second day.

Step 5 Click **Test & Save**.

Step 6 To check the health of your storage setup, click the **Diagnostics** tab and click **Run Test**.

By running a test, you can review the external storage diagnostics such as bandwidth, latency, and multiple access test details to help identify the possible storage performance issues.



CHAPTER 6

Customize Heuristic Packages

This section explains the following topics:

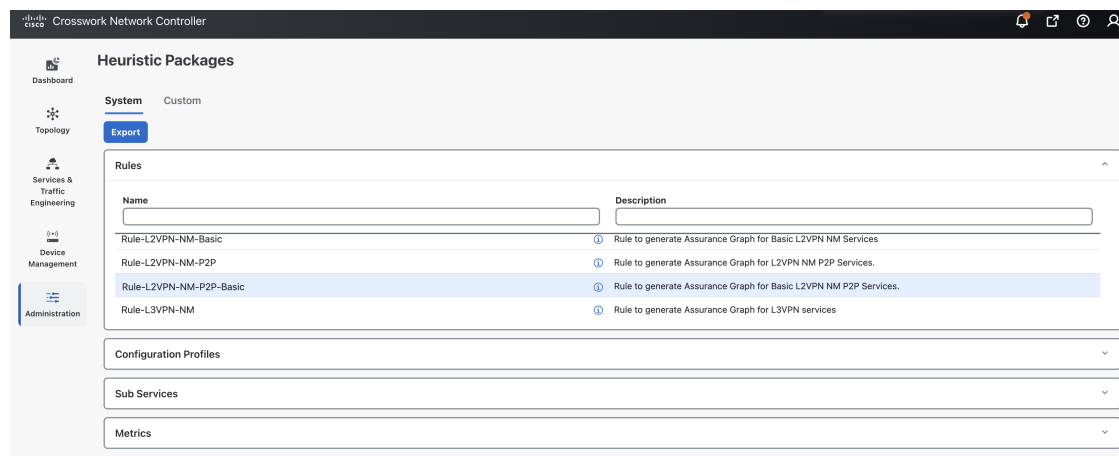
- [About Heuristic Packages, on page 49](#)
- [Build a Custom Heuristic Package, on page 51](#)
- [Import Custom Heuristic Packages, on page 53](#)

About Heuristic Packages

Crosswork Service Health uses Heuristic Packages as the core logic to monitor and report the health of services. Heuristic Packages define a list of rules, configuration profiles, supported subservices and associated metrics for every service type.

To access the Heuristic Packages, from the Main Menu, choose **Administration** > **Heuristic Packages**. The **Heuristic Packages** page has two tabs - **System** and **Custom**. The default set of Heuristic packages provided with Service Health are called system packages. These packages are available in the **System** tab. System packages cannot be modified. To customize a package to match your preferences you need to export, modify, and then import it back as a custom package. You can view the custom packages in the **Custom** tab.

Expand each section in this page to get more details on the services monitored and the thresholds used to generate alerts. You can also hover your mouse over the information **i** icon for finer details and definitions.



- **Rules:** Rules are used to structure services and the dependant sub-services and metrics within a specific service type. Dependencies within these rules help define the sub-services and the metrics that will

required for generating the data to assess the health of the service. A service can depend on an individual sub-service, a list of sub-services of the same type, or sub-services of different types.

For list of rules supported in Service Health, see [Basic and Advanced Monitoring Rules, on page 55](#).

- **Configuration Profiles:** Configuration Profiles define threshold values that act as benchmarks for assessing the health of the service. By setting specific threshold values, Configuration Profiles establish the criteria for determining when a monitored parameter is within an acceptable range or deviates from the norm.

Service Health with system heuristics package includes two configuration profiles - Silver and Gold for each of the service types (L2VPN and L3VPN). You can choose a profile option that aligns with your specific monitoring requirements. For instance, a Silver profile has more lenient thresholds compared to a Gold profile. You can create custom configuration profiles as needed.

- **Sub Services:** Sub services are characterized by a list of metrics to fetch and a list of computations to apply to these metrics in order to produce a health status and associated symptoms for the service.

For example, the sub-service *subservice.evpn.health* monitors EVPN health. It is dependant on the metric *metric.l2vpn.xconnect.pwn.state*. It evaluates an expression to check if *evpn_state* is **Up** and raises a symptom if degraded.

For list of sub-services supported in Service Health, see [Reference - Supported Subservices, on page 79](#).

- **Metrics:** Metrics define the operational data that should be fetched from different device types. Service Health uses a metric engine to map device-independent metrics to device-specific implementations, supporting multiple combinations of platforms and operating systems.

For example, fetching the metric *resource.cpu* depends on the device type. For Cisco IOS XR devices, it uses Model-Driven Telemetry (MDT), while for Cisco IOS XE devices, it relies on CLI scraping using the command `show platform resources`.

In essence, there is a hierarchical relationship between Rules, Configuration Profiles, Sub services, and Metrics. Specifically, each Rule is mapped to a type of service, and depends on a number of sub-services to compute service health, sub-services use metrics and configuration profiles set threshold values for the metrics. Based on the values defined in these files, Service Health assesses the health of the service and builds the Assurance Graph.

Here is an [Example](#) that illustrates the hierarchical relationship between Rules, Sub services, and Metrics.

Customizing Heuristic Packages

The Heuristic Package (HP) bundled with Service Health functions as an assurance model for monitoring L2VPN and L3VPN services. However, the configurations of underlay and overlay networking services may vary across deployments. While the Heuristic Package can adapt automatically to certain configuration variations, other variations cannot be seamlessly absorbed. Examples of such variations include changes in the service function pack model, the introduction of a new device type in VPN service deployment, or the introduction of new network features requiring monitoring. In these scenarios, customization may be necessary for configuration profiles, rules, metrics, or sub-service class definitions.

Refer to the section [Build a Custom Heuristic Package, on page 51](#) for a basic example of how you can create a custom package by customizing the configuration profile for a service. For further details and assistance in building a custom package based on rules, metrics, or sub-services, reach out to Cisco's Customer Experience (CX) team or your Cisco account team.

Build a Custom Heuristic Package

The procedure outlined below provides the steps for building a custom Heuristic Package by adjusting the threshold for acceptable CPU usage on the device (`CPU_THRESHOLD_MAX`) within the `Gold_L2VPN_ConfigProfile` of the Heuristic Package for L2VPN services.

Step 1 From the main menu, choose **Administration** > **Heuristic Packages**. The Heuristic Packages page opens with **System** and **Custom** tabs.

Step 2 Click the **System** tab and then click **Export**.

The `exportAPI.tar.gz` package gets downloaded to your system.

Step 3 Untar the `exportAPI.tar.gz` file, and you will get a `system` folder.

Step 4 In the `ConfigProfile` folder, open the `Gold_L2VPN_ConfigProfile-system.json` file.

Step 5 Make the following changes in the file:

- a) Change the `namespace` attribute for the profile to `custom`.
- b) Search for `CPU_THRESHOLD_MAX` and update the threshold value to 80.

```
{
  "name": "Gold_L2VPN_ConfigProfile",
  "namespace": "custom",
  "version": "1.0",
  "description": "Thresholds to use for Gold L2VPN services",
  "rules": [
    {
      "name": "Rule-L2VPN-NM",
      "namespace": "system"
    },
    {
      "name": "Rule-L2VPN-NM-P2P",
      "namespace": "system"
    },
    {
      "name": "Rule-L2VPN-NM-Basic",
      "namespace": "system"
    },
    {
      "name": "Rule-L2VPN-NM-P2P-Basic",
      "namespace": "system"
    },
    {
      "name": "Rule-L2VPN-MP-Basic",
      "namespace": "system"
    },
    {
      "name": "Rule-L2VPN-MP",
      "namespace": "system"
    }
  ],
  "values": {
    "MAX_ACCEPTABLE_IN_OUT_PKT_DELTA": {
      "description": "Max allowed difference between packets received and packets transmitted",
      "type": "VAL_INT",
      "intVal": {
        "unit": "NA",
        "val": 100
      }
    }
  }
}
```

```

    },
    "VPN_INTF_PKT_ERROR_THRESHOLD": {
      "description": "Acceptable delta of in(or out) packet errors expected between polling intervals",

      "type": "VAL_INT",
      "intVal": {
        "unit": "NA",
        "val": 10
      }
    },
    "VPN_INTF_PKT_DISCARDS_THRESHOLD": {
      "description": "Acceptable delta of in(or out) packet discards expected between polling
intervals",
      "type": "VAL_INT",
      "intVal": {
        "unit": "NA",
        "val": 10
      }
    },
    "LATENCY_RT_THRESHOLD": {
      "description": "High Threshold for latency health checks",
      "type": "VAL_INT",
      "intVal": {
        "unit": "MSEC",
        "val": 500
      }
    },
    "JITTER_RT_THRESHOLD": {
      "description": "Threshold for acceptable jitter",
      "type": "VAL_FLOAT",
      "floatVal": {
        "unit": "MSEC",
        "val": 80
      }
    },
    "PACKET_LOSS_THRESHOLD": {
      "description": "Threshold for acceptable packet loss rate",
      "type": "VAL_FLOAT",
      "floatVal": {
        "unit": "PERCENT",
        "val": 1
      }
    },
    "SRPM_DELAY_THRESHOLD": {
      "description": "High Threshold for SR-PM latency health checks",
      "type": "VAL_INT",
      "intVal": {
        "unit": "MSEC",
        "val": 200
      }
    },
    "CPU_THRESHOLD_MAX": {
      "description": "Threshold for acceptable CPU usage on the device.",
      "type": "VAL_FLOAT",
      "floatVal": {
        "unit": "PERCENT",
        "val": 80
      }
    },
    "MEMFREE_THRESHOLD_MIN": {
      "description": "Threshold for minimum free memory to be available on the device.",
      "type": "VAL_FLOAT",
      "floatVal": {

```



```

        "unit": "BYTES",
        "val": 2000000000
    }
}
}
}

```

Step 6 Save the file once you have finished making the changes.

Step 7 Create a compressed tar.gz file from the `system` folder.

What to do next

Import the custom Heuristic Package in Crosswork Network Controller. See [Import Custom Heuristic Packages, on page 53](#).

Import Custom Heuristic Packages


Follow this procedure to import the custom heuristic package in Crosswork Network Controller.

Step 1 From the main menu, choose **Administration > Heuristic Packages**. The Heuristic Packages page opens with **System** and **Custom** tabs.

Step 2 Click the **Custom** tab and then click **Import**. The **Import Heuristic Packages** dialog box appears.

Step 3 Click **Browse** to locate the custom package (*.tar.gz file) on your system.

Import Heuristic Package

 System performance might be impacted during Heuristic package import due to high server resource consumption.

File Name

Browse

Preview



I acknowledge that after importing heuristic packages, already monitored services cannot be edited/paused. Existing monitored services need to be restarted for new Heuristic package to take into effect.

Cancel

Import

Step 4 Select your custom package and click **Preview** to review the details of the package to be imported. Further information on the package's Rules, Configuration Profiles, Sub Services, and Metrics appears.

Note Your system performance might be impacted during heuristic package import due to high server resource consumption.

Select each option to preview the details of the custom package. Crosswork Network Controller will validate the package and display an error message if any issues exist. If there are no validation errors, Crosswork Network Controller will display a success message.

Step 5 Select the check box to acknowledge the warning and click **Import**. The package gets imported in Crosswork Network Controller and appears in the **Custom** tab in the **Configuration Profiles** section.

The screenshot shows the 'Custom' tab in the 'Configuration Profiles' section. The 'Rules' section is empty. The 'Configuration Profiles' section contains two entries: 'Silver_L3VPN_ConfigP...' and 'Gold_L3VPN_ConfigPr...'. The 'Gold_L3VPN_ConfigPr...' entry is highlighted with a red box.

| Name | Description |
|-------------------------|---|
| Silver_L3VPN_ConfigP... | Thresholds to use for Silver L3VPN services |
| Gold_L3VPN_ConfigPr... | Thresholds to use for Gold L3VPN services |

What to do next

To monitor services with custom heuristic packages, stop monitoring the service first. Start monitoring the service again by selecting the custom package and click **Start Monitoring**. See Step 4 in the procedure [Start Service Health monitoring, on page 11](#) for more information.



APPENDIX A

Reference - Basic Monitoring and Advanced Monitoring Rules

This section explains the following topics:

- [Basic and Advanced Monitoring Rules, on page 55](#)

Basic and Advanced Monitoring Rules

Crosswork Service Health monitoring provides two options for monitoring: Basic Monitoring and Advanced Monitoring. The table below outlines the monitoring functions of each rule and sub-services, as well as the metric dependencies for both Basic and Advanced monitoring rules included in the system-defined Heuristic Package:

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|---------------------|--|---|
| Rule-L2VPN-NM-Basic | <ul style="list-style-type: none">• Checks the health of the VPWS xconnect state.• Monitors the health of the device: CPU and memory utilization. | subservice.device.health subservice.vpws.ctrlplane.health metric.l2vpn.xconnect.state metric.l2vpn.xconnect.ac.state metric.l2vpn.xconnect.pw.state |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|-----------------------------|---------------------------------|--------------------------------|
| Rule-L2VPN-NM (Advanced) | | |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|------------------|--|---|
| | <ul style="list-style-type: none"> • Checks the health of the VPWS or EVPN xconnect state. • Monitors the health of the device: CPU and memory utilization. • Monitors the delta between received and transmitted packets between VPN interfaces and Pseudo-wire. • Monitors Y.1731 probe stats for jitter, loss, and delay metrics, and compares against SLA thresholds. • Monitors the health status of RSVP tunnel. Subservice health will be marked as 'degraded' in either of the below scenarios: <ul style="list-style-type: none"> • FRR is configured, but backup is not ready. • FRR backup is active (primary failed and traffic is flowing over FRR backup). • Health check for interface metrics: Oper status, interface in/out error packets, interface in/out packet discard. • Checks BGP Neighbor session health. • Checks whether all BGP EVPN next hops for a given L2VPN service are reachable over LSP. • Monitors PCEP session state to all the peers configured on this device. • Checks path reachability between two endpoints. • SR Policy (PCC initiated) | subservice.bgp.nbr.health subservice.bgp.evpn.nexthop.health subservice.device.health subservice.evpn.health (one for each endpoint) subservice.fallback.path.health subservice.interface.health (one for each interface) subservice.l2vpn.y1731.health subservice.path.reachability.to.peer (local to remote and remote to local) subservice.path.sla subservice.pcep.session.health (one for each endpoint device) subservice.plain.lsp.path.health subservice.sr.policy.pce.health (one for each endpoint) subservice.vpws.ctrlplane.health (local, remote) subservice.path.reachability.to.peer subservice.fallback.path.health subservice.mpls.rsvpte.tunnel.pm.health subservice.l2vpn.y1731.health subservice.vpws.ctrlplane.health subservice.interface.health subservice.device.health subservice.interface.health.summary subservice.path.sla.summary metric.bgp.router.id metric.cef.route.labeled.lsp metric.l2vpn.xconnect.ac.state metric.l2vpn.xconnect.pw.state metric.l2vpn.xconnect.state metric.device.xconnect.ac.in.packets metric.device.xconnect.pw.out.packet metric.l2vpn.y1731.connect.cross.check.status metric.interface.oper |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|------------------------------|--|--|
| | <p>health status. Admin should be up. Oper should be up. Oper should have stayed up since last polling.</p> <ul style="list-style-type: none"> Checks whether LSP path exists (in default VRF) towards the given destination device. | <p>metric.interface.in.errors metric.device.cpu.load metric.device.memory.free</p> |
| Rule-L2VPN-NM-P2P-Basic | <ul style="list-style-type: none"> Checks the health of the VPWS xconnect state. Monitors the health of the device: CPU and memory utilization. | <p>subservice.device.health subservice.vpws.ctrlplane.health</p> |
| Rule-L2VPN-NM-P2P (Advanced) | <ul style="list-style-type: none"> Checks the health of the VPWS xconnect state. Monitors the health of the device: CPU and memory utilization. Checks the health for interface metrics: Oper status, interface in/out error packets, interface in/out packet discard. Monitors Y.1731 probe stats for jitter, loss, and delay metrics, and compares against SLA thresholds. Monitors the LSP path to the peer VPN node. Monitors path reachability between two endpoints. Monitors LSP path (in default VRF) towards the given destination IP address. Monitors PCEP session state to all the peers configured on this device. Checks the SR Policy (PCC initiated) health status. Admin should be up. Oper should be up. Oper should have stayed up since last polling. | <p>subservice.device.health subservice.interface.health (one for each interface) subservice.l2vpn.y1731.health subservice.p2p.fallback.path.health subservice.p2p.path.reachability.to.peer (path reachability between endpoints) subservice.p2p.plain.lsp.path.health subservice.path.sla subservice.pcep.session.health (one for each endpoint device) subservice.sr.policy.pcc.health subservice.sr.policy.pce.health (one for each endpoint) subservice.vpws.ctrlplane.health (local, remote) metric.cef.route.labeled.lsp metric.l2vpn.xconnect.ac.state metric.l2vpn.xconnect.pw.state metric.l2vpn.xconnect.state</p> |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|---------------------|--|---|
| Rule-L2VPN-MP-Basic | <ul style="list-style-type: none">• For all .summary subservices: Groups together all the device subservices as an aggregator node. It does not have its own health/metric. Its health depends on its child subservice health.• Monitors the health of the device• Monitors bridge domain state on a given endpoint. | subservice.device.summary subservice.bridge.domain.summary subservice.device.health subservice.bridge.domain.state |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|-----------------------------|--------------------------|-------------------------|
| Rule-L2VPN-MP (Advanced) | | |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|------------------|---|---|
| | <ul style="list-style-type: none"> • For all .summary subservices: Groups together all the device subservices as an aggregator node. It does not have its own health/metric. Its health depends on its child subservice health. • Monitors the health of the device. • Groups together all the PCEP session health subservices. • Monitors PCEP session state to all the peers configured on this device. • Groups together all the device subservices. • Checks BGP Neighbor health. • Monitors whether any routes are present for the given Bridge Domain. • Groups together all the bridge domain subservices. • Monitors bridge domain state on a given endpoint. • Subservice to reflect interface health. • Groups together all the transport subservices. • SR Policy health status reflecting SR-PM SLA (if configured). Admin and Oper should be up. Oper should have stayed up since last polling. Delay and Variance should meet SLA if SR-PM is configured to measure delay. Liveness should be up if SR-PM is configured for Liveness. • Monitors the policies deployed by the ODN. | <p>subservice.device.summary</p> <p>subservice.device.health</p> <p>subservice.pcep.session.health.summary</p> <p>subservice.pcep.session.health</p> <p>subservice.evpn.summary</p> <p>subservice.bgp.nbr.health</p> <p>subservice.mac.learning</p> <p>subservice.bridge.domain.summary</p> <p>subservice.bridge.domain.state</p> <p>subservice.interface.health</p> <p>subservice.transport.summary</p> <p>subservice.sr.policy.pcc.pm.health</p> <p>subservice.sr.policy.pce.pm.health</p> <p>subservice.mpls.rsvpte.tunnel.pm.health</p> <p>subservice.l2vpn.sr.odn.policy.dynamic</p> <p>metric.device.memory.free (supports XR only)</p> <p>metric.device.cpu.load (supports XR only)</p> <p>metric.sr.te.pcc.peer.state (supports XR only)</p> <p>metric.sr.te.pcc.peer.addrs (supports XR only)</p> <p>metric.bgp.session.state (supports XR only)</p> <p>metric.bgp.neighbors.ipaddr.list (supports XR only)</p> <p>metric.mac.learning.nexthops (supports XR only)</p> <p>metric.l2vpn.bridge.ac.state (supports XR only)</p> <p>metric.l2vpn.bridge.ac.list (supports XR only)</p> <p>metric.l2vpn.bridge.domain.state (supports XR only)</p> <p>metric.interface.oper (supports both XR and XE)</p> <p>metric.interface.in.errors (supports both XR and XE)</p> <p>metric.interface.out.errors (supports both XR and XE)</p> <p>metric.interface.in.discards (supports both XR and XE)</p> |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|------------------|---|---|
| | <ul style="list-style-type: none"> • SR Policy health status that include SR-PM. Admin and Oper should be up, and Oper should have stayed up since last polling. Delay and Variance should meet SLA if SR-PM is configured to measure delay. Liveness should be up if SR-PM is configured for Liveness. • Monitors MPLS RSVP TE Tunnel Health. Admin, Oper should both be up and if FRR is configured, then backup path should be ready to pickup traffic when primary fails. If failover already happened to backup then health will be shown as degraded as there is no more redundancy in play. Delay should be considered if SR-PM is enabled. If delay is enabled, then variance will be considered. | <p>metric.interface.out.discards (supports both XR and XE)</p> <p>metric.sr.policy.pcc.admin.state (supports XR only)</p> <p>metric.sr.policy.pcc.oper.state (supports XR only)</p> <p>metric.sr.policy.pcc.oper.up.time (supports XR only)</p> <p>metric.sr.policy.pm.delay.measurement (supports XR only)</p> <p>metric.sr.pm.delay (supports XR only)</p> <p>metric.sr.pm.variance (supports XR only)</p> <p>metric.sr.policy.pm.liveness.detection (supports XR only)</p> <p>metric.sr.pm.liveness.state (supports XR only)</p> <p>metric.sr.policy.pcc.admin.state (supports XR only)</p> <p>metric.sr.policy.pcc.oper.state (supports XR only)</p> <p>metric.sr.policy.pcc.oper.up.time (supports XR only)</p> <p>metric.sr.policy.pcc.ietf.policy.name (supports XR only)</p> <p>metric.sr.policy.pm.delay.measurement (supports XR only)</p> <p>metric.sr.pm.delay (supports XR only)</p> <p>metric.sr.pm.variance (supports XR only)</p> <p>metric.sr.policy.pm.liveness.detection (supports XR only)</p> <p>metric.sr.pm.liveness.state (supports XR only)</p> <p>metric.mpls.rsvpte.tunnel.oper.state (supports XR only)</p> <p>metric.mpls.rsvpte.tunnel.admin.state (supports XR only)</p> <p>metric.mpls.rsvpte.tunnel.frr.configured (supports XR only)</p> <p>metric.mpls.rsvpte.tunnel.frr.status (supports XR only)</p> |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|---------------------|---|---|
| | | metric.mpls.te.pm.delay.measurement (supports XR only) metric.mpls.rsvp.te.delay (supports XR only) metric.mpls.rsvp.te.variance (supports XR only) metric.l2vpn.odn.sr.policies.list (supports XR only) metric.bgp.router.id (supports both XR and XE) |
| Rule-L3VPN-NM-Basic | <ul style="list-style-type: none"> • Reports the overall route connectivity health between the current PE device and its connecting CE device. • Monitors the health of the device: CPU and memory utilization. | subservice.ce.pe.route.health subservice.device.health |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|--------------------------|---|-------------------------|
| Rule-L3VPN-NM (Advanced) | <ul style="list-style-type: none"> • For all .summary subservices: Groups together all the device subservices as an aggregator node. It does not have its own health/metric. Its health depends on its child subservice health. • Subservice, together with child subservices in L3VPN Rule, reports the overall route health between current PE device and its connecting CE device. • eBGP Session health • Subservice to reflect interface health. • Monitors the health of the device. • L3VPN Aggregator Subservice that reflects path reachability from given device, for a given vrf, to peer VPN sites. • Monitors both static and dynamically initiated policy. • Checks whether plain LSP route exists within given VRF towards given vpn ip-addresses. • Monitors PCEP session state to all the peers configured on this device. • Checks BGP Neighbor health. | |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|------------------|--------------------------|--|
| | | subservice.ce.pe.route.health.summary subservice.ce.pe.route.health subservice.ebgp.nbr.health subservice.interface.health.summary subservice.interface.health subservice.device.summary subservice.device.health subservice.vrf.path.reachability.to.peer.summary subservice.vrf.path.reachability.to.peers subservice.transport.summary subservice.dynamic.l3vpn.sr.policy subservice.vrf.plain.lsp.reachability subservice.pcep.session.health.summary subservice.pcep.session.health subservice.bgp.nbr.health.summary subservice.bgp.nbr.health subservice.bgp.evpn.nextthop.health subservice.bgp.nbr.health subservice.ce.pe.route.health subservice.device.health subservice.ebgp.nbr.health subservice.evpn.health subservice.fallback.path.health subservice.interface.health subservice.l2vpn.y1731.health subservice.p2p.fallback.path.health subservice.p2p.path.reachability.to.peer subservice.p2p.plain.lsp.path.health subservice.path.reachability.to.peer subservice.path.sla subservice.pcep.session.health subservice.plain.lsp.path.health subservice.sr.policy.pcc.health |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|------------------|--------------------------|---|
| | | subservice.sr.policy.pce.health subservice.vpws.ctrlplane.health subservice.vrf.path.reachability.to.peers subservice.vrf.plain.lsp.reachability subservice.bridge.domain.summary subservice.l3vpn.sr.odn.policy.dynamic subservice.l2vpn.sr.odn.policy.dynamic subservice.mac.learning subservice.mpls.rsvp.tunnel.pm.health subservice.vrf.path.reachability.to.peer.summary subservice.path.sla.summary subservice.pcep.session.health.summary subservice.transport.summary subservice.interface.health.summary subservice.vpws.ctrlplane.health.summary subservice.bridge.domain.state metric.route.vrf.connected (supports XR and XR IPv6) metric.route.vrf.local (supports XR and XR IPv6) metric.bgp.vrf.session.state (supports XR only) metric.interface.oper (supports both XR and XE) metric.interface.in.errors (supports both XR and XE) metric.interface.out.errors (supports both XR and XE) metric.interface.in.discards (supports both XR and XE) metric.interface.out.discards (supports both XR and XE) metric.device.memory.free (supports XR only) metric.device.cpu.load (supports XR only) metric.l3vpn.sr.policies.list (supports XR and XR IPv6) |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|------------------|--------------------------|---|
| | | metric.cef.vrf.route.prefix (supports XR and XR IPv6) metric.sr.te.pcc.peer.state (supports XR only) metric.sr.te.pcc.peer.addrs (supports XR only) metric.bgp.session.state (supports XR only) metric.bgp.neighbors.ipaddr.list (supports XR only) metric.bgp.route.l2vpn.evpn.nexthops metric.bgp.router.id metric.cef.route.labeled.lsp metric.bgp.session.state metric.bgp.neighbors.ipaddr.list metric.route.vrf.connected metric.route.vrf.local metric.device.memory.free metric.device.cpu.load metric.bgp.vrf.session.state metric.l2vpn.xconnect.pw.state metric.cef.route.labeled.lsp metric.bgp.router.id metric.interface.oper metric.interface.in.errors metric.interface.out.errors metric.interface.in.discards metric.interface.out.discards metric.l2vpn.y1731.connect.cross.check.status metric.l2vpn.y1731.connect.peer.mep.status metric.l2vpn.y1731.latency.rt metric.l2vpn.y1731.jitter.rt metric.l2vpn.y1731.pktloss.lway.sd metric.l2vpn.y1731.pktloss.lway.ds metric.cef.route.labeled.lsp metric.cef.route.labeled.lsp metric.device.xconnect.ac.in.packets |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|------------------|--------------------------|---|
| | | metric.device.xconnect.pw.out.packets metric.device.xconnect.pw.in.packets metric.device.xconnect.ac.out.packets metric.sr.te.pcc.ipv4.peer.state metric.sr.te.pcc.ipv4.peer.addr metric.cef.route.labeled.lsp metric.bgp.router.id metric.sr.policy.pcc.oper.state metric.sr.policy.pcc.oper.up.time metric.sr.policy.pcc.admin.state metric.sr.policy.pm.delay.measurement metric.sr.pm.delay metric.sr.pm.variance metric.sr.policy.pm.liveness.detection metric.sr.pm.liveness.state metric.sr.policy.pce.oper.up.time metric.sr.policy.pce.oper.state metric.sr.policy.pce.admin.state metric.l2vpn.xconnect.state metric.l2vpn.xconnect.ac.state metric.l2vpn.xconnect.pw.state metric.cef.vrf.route.prefix metric.l3vpn.odn.sr.policies.dynamic.list metric.l2vpn.odn.sr.policies.list metric.bgp.router.id metric.mac.learning.nexthops metric.mpls.rsvpte.tunnel.oper.state metric.mpls.rsvpte.tunnel.admin.state metric.mpls.rsvpte.tunnel.frr.configured metric.mpls.rsvpte.tunnel.frr.status metric.mpls.te.pm.delay.measurement metric.mpls.rsvp.te.delay metric.l2vpn.bridge.ac.state |

| Rule Name (Type) | Monitoring Functionality | Metrics and Subservices |
|------------------|--------------------------|---|
| | | metric.l2vpn.bridge.ac.list metric.l2vpn.bridge.domain.state |

Example

The given example explains the relationship between the 'Rule-L2VPN-NM-P2P-Basic' and its dependent sub-services, specifically 'subservice.vpws.ctrlplane.health' and 'subservice.device.health'. Additionally, the sub-service definitions are also listed below to highlight the metric dependencies and symptoms generated by these sub-services.

Rule-L2VPN-NM-P2P-Basic

```
{
  "name": "Rule-L2VPN-NM-P2P-Basic",
  "namespace": "system",
  "id": "Rule-L2VPN-NM-P2P-Basic system",
  "description": "Rule to generate Assurance Graph for Basic L2VPN NM P2P Services.",
  "matchCriteria": [
    {
      "configSource": "SOURCE_TYPE_NS0",
      "configSubSource": [
        "SUBSOURCE_SERVICE_CONFIG"
      ],
      "matchType": "MATCH_TYPE_XPATH",
      "matchExpression":
        "//vpn-service[@xmlns='urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw']/vpn-svc-type[text()='vpn-common:t-ldp']",
      "matchPrefix": "",
      "matchParams": []
    },
    {
      "configSource": "SOURCE_TYPE_NS0",
      "configSubSource": [
        "SUBSOURCE_SERVICE_CONFIG"
      ],
      "matchType": "MATCH_TYPE_XPATH",
      "matchExpression":
        "//flat-L2vpn/service-type[text()='p2p']",
      "matchPrefix": "",
      "matchParams": []
    },
    {
      "configSource": "SOURCE_TYPE_NS0",
      "configSubSource": [
        "SUBSOURCE_SERVICE_CONFIG"
      ],
      "matchType": "MATCH_TYPE_XPATH",
      "matchExpression":
        "//vpn-service[@xmlns='urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw']/vpn-type[text()='vpn-common:t-ldp']",
      "matchPrefix": "",
      "matchParams": []
    },
    {
      "configSource": "SOURCE_TYPE_NS0",
      "configSubSource": [
        "SUBSOURCE_SERVICE_CONFIG"
      ],
      "matchType": "MATCH_TYPE_XPATH",
```

```

        "matchExpression": "//vpn-service[not(//bridge-group)]/vpn-type[contains(text(),
':mpls-evpn')]",
        "matchPrefix": "",
        "matchParams": []
    },
    {
        "configSource": "SOURCE_TYPE_NS0",
        "configSubSource": [
            "SUBSOURCE_SERVICE_CONFIG"
        ],
        "matchType": "MATCH_TYPE_XPATH",
        "matchExpression":
        "//vpn-service[@xmlns='urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw']/vpn-type[text()='x:vpws']",

        "matchPrefix": "",
        "matchParams": []
    },
    {
        "configSource": "SOURCE_TYPE_NS0",
        "configSubSource": [
            "SUBSOURCE_SERVICE_CONFIG"
        ],
        "matchType": "MATCH_TYPE_XPATH",
        "matchExpression":
        "//vpn-service[@xmlns='urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw']/vpn-type[text()='ietf-vpn-common:vpws']",

        "matchPrefix": "",
        "matchParams": []
    }
],
"dependencies": [
    {
        "name": "VPWS-ControlPlane-Health-Summary",
        "id": "subservice.vpws.ctrlplane.health.summary system",
        "ssClass": "subservice.vpws.ctrlplane.health.summary",
        "namespace": "system",
        "type": "DEP_TYPE_NON_LIST",
        "optional": false,
        "paramExtractionMechanism": {
            "mode": "EXTRACT_MODE_XPATH",
            "name": "",
            "namespace": "",
            "version": "",
            "validationHash": "0",
            "pluginMethod": "",
            "extractedParams": [],
            "nativeMethod": ""
        },
        "parameters": [
            {
                "name": "vpnServiceId",
                "iterator": false,
                "defaultValue": "",
                "extractionMethod": "DEP_PARAM_XPATH",
                "extractionDetails": [
                    {
                        "description": "",
                        "extractValue": "//vpn-service/vpn-id"
                    },
                    {
                        "description": "Flat Model",
                        "extractValue": "//flat-L2vpn[/flat-L2vpn-p2p]/key"
                    }
                ]
            }
        ]
    }
]

```

```

    }
  ],
  "subDependencies": [
    "VPWS-ControlPlane-Health-Local-Site",
    "VPWS-ControlPlane-Health-Remote-Site"
  ],
  "softSubDependencies": []
},
{
  "name": "VPWS-ControlPlane-Health-Local-Site",
  "id": "subservice.vpws.ctrlplane.health system",
  "ssClass": "subservice.vpws.ctrlplane.health",
  "namespace": "system",
  "type": "DEP_TYPE_NON_LIST",
  "optional": false,
  "paramExtractionMechanism": {
    "mode": "EXTRACT_MODE_XPATH",
    "name": "",
    "namespace": "",
    "version": "",
    "validationHash": "0",
    "pluginMethod": "",
    "extractedParams": [],
    "nativeMethod": ""
  },
  "parameters": [
    {
      "name": "device",
      "iterator": false,
      "defaultValue": "",
      "extractionMethod": "DEP_PARAM_XPATH",
      "extractionDetails": [
        {
          "description": "",
          "extractValue": "//vpn-nodes/vpn-node[1]/vpn-node-id"
        }
      ]
    },
    {
      "name": "groupName",
      "iterator": false,
      "defaultValue": "",
      "extractionMethod": "DEP_PARAM_XPATH",
      "extractionDetails": [
        {
          "description": "",
          "extractValue": "//vpn-service/vpn-id"
        },
        {
          "description": "Flat Model",
          "extractValue": "//flat-L2vpn/flat-L2vpn-p2p/local-site/xconnect-group-name"
        }
      ]
    },
    {
      "name": "xconnectName",
      "iterator": false,
      "defaultValue": "",
      "extractionMethod": "DEP_PARAM_XPATH",
      "extractionDetails": [
        {
          "description": "",
          "extractValue": "//vpn-service/vpn-id"
        }
      ]
    }
  ]
}

```

```

        {
          "description": "Flat Model",
          "extractValue": "//flat-L2vpn/flat-L2vpn-p2p/local-site/xconnect-group-name"
        }
      ]
    },
    "subDependencies": [],
    "softSubDependencies": [
      "device1"
    ]
  },
  {
    "name": "VPWS-ControlPlane-Health-Remote-Site",
    "id": "subservice.vpws.ctrlplane.health system",
    "ssClass": "subservice.vpws.ctrlplane.health",
    "namespace": "system",
    "type": "DEP_TYPE_NON_LIST",
    "optional": false,
    "paramExtractionMechanism": {
      "mode": "EXTRACT_MODE_XPATH",
      "name": "",
      "namespace": "",
      "version": "",
      "validationHash": "0",
      "pluginMethod": "",
      "extractedParams": [],
      "nativeMethod": ""
    },
    "parameters": [
      {
        "name": "device",
        "iterator": false,
        "defaultValue": "",
        "extractionMethod": "DEP_PARAM_XPATH",
        "extractionDetails": [
          {
            "description": "",
            "extractValue": "//vpn-nodes/vpn-node[2]/vpn-node-id"
          }
        ]
      },
      {
        "name": "groupName",
        "iterator": false,
        "defaultValue": "",
        "extractionMethod": "DEP_PARAM_XPATH",
        "extractionDetails": [
          {
            "description": "",
            "extractValue": "//vpn-service/vpn-id"
          },
          {
            "description": "Flat Model",
            "extractValue": "//flat-L2vpn/flat-L2vpn-p2p/remote-site/xconnect-group-name"
          }
        ]
      },
      {
        "name": "xconnectName",
        "iterator": false,
        "defaultValue": "",
        "extractionMethod": "DEP_PARAM_XPATH",

```

```

    "extractionDetails": [
      {
        "description": "",
        "extractValue": "//vpn-service/vpn-id"
      },
      {
        "description": "Flat Model",
        "extractValue": "//flat-L2vpn/flat-L2vpn-p2p/remote-site/xconnect-group-name"
      }
    ]
  },
  "subDependencies": [],
  "softSubDependencies": [
    "device2"
  ]
},
{
  "name": "device1",
  "id": "subservice.device.health system",
  "ssClass": "subservice.device.health",
  "namespace": "system",
  "type": "DEP_TYPE_NON_LIST",
  "optional": false,
  "paramExtractionMechanism": {
    "mode": "EXTRACT_MODE_XPATH",
    "name": "",
    "namespace": "",
    "version": "",
    "validationHash": "0",
    "pluginMethod": "",
    "extractedParams": [],
    "nativeMethod": ""
  },
  "parameters": [
    {
      "name": "device",
      "iterator": false,
      "defaultValue": "",
      "extractionMethod": "DEP_PARAM_XPATH",
      "extractionDetails": [
        {
          "description": "",
          "extractValue": "//vpn-nodes/vpn-node[1]/vpn-node-id"
        }
      ]
    }
  ],
  "subDependencies": [],
  "softSubDependencies": []
},
{
  "name": "device2",
  "id": "subservice.device.health system",
  "ssClass": "subservice.device.health",
  "namespace": "system",
  "type": "DEP_TYPE_NON_LIST",
  "optional": false,
  "paramExtractionMechanism": {
    "mode": "EXTRACT_MODE_XPATH",
    "name": "",
    "namespace": "",
    "version": "",

```

```

        "validationHash": "0",
        "pluginMethod": "",
        "extractedParams": [],
        "nativeMethod": ""
    },
    "parameters": [
        {
            "name": "device",
            "iterator": false,
            "defaultValue": "",
            "extractionMethod": "DEP_PARAM_XPATH",
            "extractionDetails": [
                {
                    "description": "",
                    "extractValue": "//vpn-nodes/vpn-node[2]/vpn-node-id"
                }
            ]
        }
    ],
    "subDependencies": [],
    "softSubDependencies": []
}
},
"softRootDependencies": [],
"createTimestamp": "1697841637567500247",
"updateTimestamp": "0",
"monitoringType": "BASIC",
"version": "1.1"
}

Sub service: 'subservice.vpws.ctrlplane.health'

{
    "id": "subservice.vpws.ctrlplane.health.summary system",
    "name": "subservice.vpws.ctrlplane.health.summary",
    "namespace": "system",
    "description": "Groups together all the VPWS Ctrlplane health subservices.",
    "params": [
        {
            "name": "vpnServiceId",
            "description": "",
            "type": "PARAM_TYPE_NON_LIST"
        }
    ],
    "liveMetrics": {},
    "rootExpressions": [],
    "dynamicConfig": null,
    "symptom": null,
    "dependencies": [],
    "exprCid": "",
    "createTimestamp": "1697841637373426164",
    "updateTimestamp": "0",
    "tags": [],
    "version": "1.0"
}

{
    "id": "subservice.vpws.ctrlplane.health system",
    "name": "subservice.vpws.ctrlplane.health",
    "namespace": "system",
    "description": "check the health of the VPWS state",
    "params": [
        {
            "name": "device",
            "description": "",

```

```

        "type": "PARAM_TYPE_NON_LIST"
    },
    {
        "name": "groupName",
        "description": "",
        "type": "PARAM_TYPE_NON_LIST"
    },
    {
        "name": "xconnectName",
        "description": "",
        "type": "PARAM_TYPE_NON_LIST"
    }
],
"liveMetrics": {},
"rootExpressions": [
    {
        "evalExpression": "xconnect_state == 'up' && ac_state == 'up' && evpn_state == 'up'",
        "activateCondition": ""
    }
],
"dynamicConfig": null,
"symptom": {
    "formatString": "VPWS State degraded. Device: {device}, XConnectGroup: {groupName},
XconnectName: {xconnectName}",
    "level": "DEGRADED",
    "priority": 15,
    "condition": false
},
"dependencies": [
    {
        "type": "DEP_TYPE_METRIC",
        "label": "xconnect_state",
        "evalExpression": "metric.l2vpn.xconnect.state",
        "namespace": "",
        "symptom": null,
        "paramMap": {
            "device": "device",
            "groupName": "groupName",
            "xconnectName": "xconnectName"
        },
        "id": ""
    },
    {
        "type": "DEP_TYPE_METRIC",
        "label": "ac_state",
        "evalExpression": "metric.l2vpn.xconnect.ac.state",
        "namespace": "",
        "symptom": null,
        "paramMap": {
            "device": "device",
            "groupName": "groupName",
            "xconnectName": "xconnectName"
        },
        "id": ""
    },
    {
        "type": "DEP_TYPE_METRIC",
        "label": "evpn_state",
        "evalExpression": "metric.l2vpn.xconnect.pw.state",
        "namespace": "",
        "symptom": null,
        "paramMap": {
            "device": "device",

```

```

        "groupName": "groupName",
        "xconnectName": "xconnectName"
    },
    "id": ""
}
],
"exprCid": "",
"createTimestamp": "1697841637370064741",
"updateTimestamp": "0",
"tags": [],
"version": "1.0"
}

```

Sub service: 'subservice.device.health'

```

{
    "id": "subservice.device.health system",
    "name": "subservice.device.health",
    "namespace": "system",
    "description": "Monitor the health of the device.",
    "params": [
        {
            "name": "device",
            "description": "",
            "type": "PARAM_TYPE_NON_LIST"
        }
    ],
    "liveMetrics": {},
    "rootExpressions": [
        {
            "evalExpression": "cpu_healthy && memory_healthy",
            "activateCondition": ""
        }
    ],
    "dynamicConfig": null,
    "symptom": {
        "formatString": "Heavier than expected resource consumption on the Device: {device}",
        "level": "DEGRADED",
        "priority": 100,
        "condition": false
    },
    "dependencies": [
        {
            "type": "DEP_TYPE_EXPRESSION",
            "label": "cpu_healthy",
            "evalExpression": "ListElemsAverage(cpu_load) <= CPU_THRESHOLD_MAX",
            "namespace": "",
            "symptom": null,
            "paramMap": {},
            "id": ""
        },
        {
            "type": "DEP_TYPE_EXPRESSION",
            "label": "memory_healthy",
            "evalExpression": "ListElemsSum(memory_free) > MEMFREE_THRESHOLD_MIN",
            "namespace": "",
            "symptom": null,
            "paramMap": {},
            "id": ""
        },
        {
            "type": "DEP_TYPE_METRIC",
            "label": "cpu_load",
            "evalExpression": "metric.device.cpu.load",
            "namespace": "",

```



```

        "symptom": null,
        "paramMap": {
          "device": "device"
        },
        "id": ""
      },
      {
        "type": "DEP_TYPE_METRIC",
        "label": "memory_free",
        "evalExpression": "metric.device.memory.free",
        "namespace": "",
        "symptom": null,
        "paramMap": {
          "device": "device"
        },
        "id": ""
      }
    ],
    "exprCid": "",
    "createTimestamp": "1697841637256704609",
    "updateTimestamp": "0",
    "tags": [
      "DEVICE_SUBSERVICES"
    ],
    "version": "1.1"
  }

  {
    "id": "subservice.device.summary system",
    "name": "subservice.device.summary",
    "namespace": "system",
    "description": "Groups together all the Device subservices",
    "params": [
      {
        "name": "vpnServiceId",
        "description": "",
        "type": "PARAM_TYPE_NON_LIST"
      }
    ],
    "liveMetrics": {},
    "rootExpressions": [],
    "dynamicConfig": null,
    "symptom": null,
    "dependencies": [],
    "exprCid": "",
    "createTimestamp": "1697841637260108075",
    "updateTimestamp": "0",
    "tags": [],
    "version": "1.0"
  }

```




APPENDIX B

Reference - Supported Subservices

The following tables provide details of supported Service Health L2VPN/L3VPN flavors and associated subservices (for IOS XE and IOS XR devices).

Table 6: Supported VPN Services with Associated Subservices (for IOS XE Devices)

| Supported VPN Services | Associated Subservices | Details |
|---------------------------------------|--|--|
| L2VPN Point to Point with SR underlay | <ul style="list-style-type: none"> • Path Reachability • Y.1731 Health • VPN Interface Health • Device Health • Summary (aggregator) nodes | IOS XE does not support SNMP/gNMI this subservice (CEF route; PCEP Session State; XConnect). |
| L2VPN Point to Point over MPLS LDP | <ul style="list-style-type: none"> • Path Reachability • Y.1731 Health • VPWS Control Plane health • VPN Interface Health • Device Health • Summary (aggregator) nodes | IOS XE does not support SNMP/gNMI this subservice (CEF route; XConnect). |
| L2VPN P2P Plain | <ul style="list-style-type: none"> • Path Reachability • Y.1731 Health • VPN Interface Health • Device Health • Summary (aggregator) nodes | IOS XE does not support SNMP/gNMI this subservice (CEF route; XConnect). Note: The reference to 'Plain' implies that traffic takes the IGP path and does not use SR. |

| | | |
|----------|---|---|
| L3VPN SR | <ul style="list-style-type: none"> • Path Reachability • CE-PE Route Health • eBGP Neighbor Health • VPN Interface Health • BGP Neighbor Health (DynExp) • Summary (aggregator) nodes | IOS XE does not support SNMP/gNMI col... this subservice (CEF route; PCEP Session S... is also not supported. |
|----------|---|---|

Table 7: Supported VPN Services with Associated Subservices (for IOS XR Devices)

| Supported VPN Services | Associated Subservices |
|------------------------|--|
| L2VPN EVPN SR | <ul style="list-style-type: none"> • Path Reachability • Fallback Enabled/Disabled (DynExp) • SR Policy – PCC • Path SLA • Y.1731 Health • VPWS Control Plane Health • VPN Interface Health • Device Health • EVPN Health • BGP Neighbor Health (DynExp) • BGP Nexthop Health (DynExp) • PCEP Session Health (DynExp) • SR Policy – PCE • Summary (aggregator) nodes |

| | |
|---------------------------------------|--|
| L2VPN EVPN Plain | <ul style="list-style-type: none"> • Path Reachability • Path SLA • Plain LSP Path Health (DynExp) • VPWS Control Plane health • VPN Interface Health • Device Health • EVPN Health • BGP Neighbor Health (DynExp) • BGP Nexthop Health (DynExp) • Summary (aggregator) nodes <p>Note: The reference to 'Plain' implies that L2VPN/L3VPN traffic takes the IGP path and does not use any transports, like SR.</p> |
| L2VPN Point to Point over RSVP | <ul style="list-style-type: none"> • Path Reachability • Fallback Enabled/Disabled • RSVP-TE Health • Path SLA • Y.1731 Health • VPWS Control Plane Health/Xconnect Health • VPN Interface Health • Device Health |
| L2VPN Point to Point with SR underlay | <ul style="list-style-type: none"> • Path Reachability • Fallback Enabled/Disabled • SR Policy – PCC • Path SLA • Y.1731 Health • VPWS Control Plane Health • VPN Interface Health • Device Health • PCEP Session Health (DynExp) • SR Policy – PCE • Summary (aggregator) nodes |

| | |
|------------------------------------|--|
| L2VPN Point to Point over MPLS LDP | <ul style="list-style-type: none"> • Path Reachability • Fallback Enabled/Disabled • Path SLA • Y.1731 Health • VPWS Control Plane Health • VPN Interface Health • Device Health • Summary (aggregator) nodes |
| L2VPN P2P Plain | <ul style="list-style-type: none"> • Path Reachability • Plain LSP Path Health • Path SLA • Y.1731 Health • VPWS Control Plane Health • VPN Interface Health • Device Health • Summary (aggregator) nodes <p>Note: The reference to ‘Plain’ implies that L2VPN/L3VPN traffic takes the IGP path and does not use any transports, like SR.</p> |
| L3VPN SR | <ul style="list-style-type: none"> • CE-PE Route Health • eBGP Neighbor Health • VPN Interface Health • Device Health • Path Reachability • Vrf Plain LSP Path Health • PCEP Session Health (DynExp) • BGP Neighbor Health (DynExp) • Summary (aggregator) nodes • SR and SRv6 polices |