Cisco Crosswork Network Controller 7.2.x Service Health Monitoring

Cisco Crosswork Network Controller
Updated January 30, 2026

© 2026 Cisco Systems, Inc. All rights reserved.

Full Cisco Trademarks with Software License

Full Cisco Trademarks with Software License

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 NONINFRINGEMENT 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)

Topics included

1 Introduction.....5

Overview of Cisco Crosswork Network Controller Service Health monitoring.....6

Audience and prerequisites.....6

Service Health.....6

Service health monitoring capabilities.....7

Service Health APIs.....7

2 Getting started.....9

Important changes in behavior.....10

Before you begin.....10

Getting started.....12

Service health monitoring workflows.....13

Managing stored data.....13

Analyzing the cause of service degradation.....14

Using SR-PM to monitor links and TE policies.....14

Monitoring service health using provider connectivity assurance.....15

Customizing Heuristic Packages.....15

Service Health audit logging.....16

View the source IP in Service Health audit logs.....16

Service Health monitoring scale information.....17

Service Health monitoring single-VM scale information.....18

3 Monitor Service Health.....19

Start Service Health monitoring.....20

Adjust monitoring settings.....21

Edit the monitoring settings.....22

Pause and resume Service Health monitoring.....23

Stop Service Health monitoring.....24

SR-PM monitoring for links and TE policies.....25

Enable SR-PM metrics collection.....26

View TE policy performance metrics.....26

View link performance metrics.....28

Service health monitoring using CS-SR policies.....28

Service health monitoring using Cisco provider connectivity assurance.....29

Add Provider Connectivity Assurance as a provider.....29

Check Provider Connectivity Assurance reachability.....31

View probe session details.....32

Historical data from probe sessions.....34

Known issues and limitations with Provider Connectivity Assurance....36

4 Analyze Service Health....37

Monitored services overview....38
VPN Services page....39
Service Health Dashboard....40
Monitoring status of services....41
Identify the active symptoms and root causes of a degraded service....43
Assurance graphs....46
Use the Assurance Graph to identify root causes....48
Identify root causes using last 24 Hr metrics....50
View the devices participating in the service....53
View the collection jobs....54

5 Configure additional storage....57

Configure the additional external storage....58

6 Customize Heuristic Packages....61

Heuristic packages....62
Create a custom heuristic package....63
Import the custom heuristic packages....65
Custom heuristic packages and rule mismatches....66
Custom heuristic packages after Crosswork Network Controller upgrades....67

Appendix A: Reference - basic monitoring and advanced monitoring Rules....69

Basic and advanced monitoring rules, subservices, and metrics....70
Service and subservice dependency example....84

Appendix B: Supported VPN services with associated subservices....95

1 Introduction

Topics:

- Overview of Cisco Crosswork Network Controller Service Health monitoring

This section explains these topics:

Overview of Cisco Crosswork Network Controller Service Health monitoring

Explains the role of the Service Health component within the Crosswork Network Controller Advantage package and summarizes its primary monitoring and assurance benefits.

Cisco Crosswork Network Controller Service Health monitoring is a component of the Crosswork Network Controller Advantage package that:

- Offers a service-level perspective that assesses the impact of physical network issues on services,
- Iidentifies specific locations and causes of service degradation, and
- Provides targeted tools for service-specific monitoring and assurance.

Audience and prerequisites

Describes the intended audience for the Service Health guide and outlines the technical familiarity and prerequisite knowledge required for network administrators.

Audience and prerequisites are documentation components that:

- Identify network administrators as the primary users of the Service Health component,
- Specify the required familiarity with platform infrastructure and L2VPN or L3VPN service provisioning, and
- List the networking technologies and traffic engineering protocols required for effective use.

Required technical familiarity

Users must be familiar with these topics:

- Platform Infrastructure and installation of Crosswork Network Controller components. For more information, refer to the Crosswork Network Controller 7.2.x Installation guide.
- Provisioning Layer 2 Virtual Private Network (L2VPN) and Layer 3 Virtual Private Network (L3VPN) services.
- Networking technologies and protocols, including Border Gateway Protocol - Link State (BGP-LS), Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS), Path Computation Element Communication Protocol (PCEP), and model-driven telemetry.
- Traffic Engineering (TE) tunnels, including Resource Reservation Protocol-Traffic Engineering (RSVP-TE) and Segment Routing Traffic Engineering (SR-TE) policy provisioning.

Service Health

Explains the conceptual framework of Service Health, detailing how it assesses service impact beyond physical link monitoring to provide targeted troubleshooting tools.

Service Health is a component of the Cisco Crosswork Network Controller Advantage package that:

- Offers a service-level perspective that goes beyond monitoring devices at a physical level,
- Assesses the impact of physical network issues on services traversing the links, such as L2 and L3 VPN services,
- Provides operators with tools for service-specific monitoring and assurance to pinpoint why a service is degraded,
- Offers APIs and samples for developing additional service monitoring tools, and
- Allows for the development of additional service monitoring capabilities to meet specific requirements.

Service health monitoring benefits

Service Health extends monitoring capabilities by offering a service-level perspective. While the controller can alert you to a failed link or one that has reached capacity, Service Health assesses how these issues affect the specific services using those links.

Service health monitoring capabilities

Provides a factual list of the network monitoring scenarios supported by Service Health, including point-to-point and multipoint L2VPN and L3VPN services.

Describe the network monitoring scenarios supported by Service Health, including point-to-point and multipoint L2VPN and L3VPN services, to help users understand how to evaluate and visualize service health across their network.

Service Health includes packages for monitoring several common network scenarios:

- Health monitoring of point-to-point L2VPN services, such as T-LDP and EVPN.
- Health monitoring of multipoint L2VPN services, including EVPN E-LAN and E-Tree L2VPN EVPN.
- Health monitoring of L3VPN services.
- Analysis of health for degraded services to provide troubleshooting information.
- Visualization of service health status and the associated logical dependency tree.

Service Health APIs

Describes the use of application programming interfaces to integrate Service Health with other controller components and third-party applications for custom tool development.

Service Health APIs are application programming interfaces that:

- Integrate other Crosswork Network Controller components with Service Health,
- Connect third-party applications to deliver new capabilities into network operations, and
- Provide the necessary documentation on Cisco DevNet for developing custom monitoring tools.

API documentation access

For more information on using the APIs, refer to the API documentation on Cisco DevNet.

2 Getting started

Topics:

- Important changes in behavior
- Before you begin
- Getting started
- Service health monitoring workflows
- Service Health audit logging
- Service Health monitoring scale information
- Service Health monitoring single-VM scale information

This section explains these topics:

Important changes in behavior

Lists the important behavioral changes in this Service Health release, including the discontinuation of AWS external storage for historical monitoring data.

Describe the changes in behavior introduced in this Service Health release, with a focus on the discontinuation of AWS external storage for historical monitoring data, so that users understand how data storage and retention policies are affected.

This release introduces these important changes in the behavior of Service Health.

Table 1: Changes in behavior

Topic	Description
Discontinuation of Amazon Web Services (AWS) external storage for Service Health historical data.	Service Health historically allowed configuration of external storage using an AWS cloud account to store monitoring data beyond the internal storage capacity. This external storage acted as a cloud-based archive, while the internal storage functioned as a local cache. Service Health no longer supports storing historical data on AWS. With this change, historical data retention is now limited to the 50 GB internal storage capacity. To prevent loss of access to older historical data, regularly observe the health of the monitored services.

Before you begin

Describes the Service Health concepts, monitoring levels, operational limits, and Heuristic Packages to familiarize yourself with before you start monitoring services.

Before you begin, familiarize yourself with the key Service Health concepts that determine how you plan and run monitoring:

- The views available for assessing the health of VPN services,
- The Basic and Advanced monitoring levels and the operational limits that apply, and
- The Heuristic Packages that define the rules used to monitor service health.

Service health views

Crosswork Network Controller provides these views for monitoring VPN service health:

- **VPN service health** dashlet: From the Crosswork Network Controller home page, this dashlet delivers an at-a-glance summary of VPN service statuses and enables visibility and filtering for monitoring status, such as errors.
- **VPN Services** page: This page presents detailed health and monitoring data for all VPN services, enhanced by customizable columns and filters that support focused troubleshooting.
- **Service Health Dashboard**: This dashboard aggregates service metrics and highlights SLA breaches, enabling efficient oversight of network performance and service quality.

Monitoring levels

Crosswork Network Controller monitors services at two levels:

- **Basic Monitoring**: Monitors a higher number of services and provides limited subservice metrics, resulting in lower resource consumption and smaller graphic map renderings.
- **Advanced Monitoring**: Supported for a fewer number of services because it monitors a larger number of component subservices and consumes more compute resources, resulting in an increased number of subservice metrics and larger graphic map renderings.

To view only **Basic** or **Advanced** services from the Crosswork Network Controller home page, click the highlighted number within the dashlet. The VPN Services page appears with the filtered service information.

Crosswork Network Controller Service Health supports single virtual machine (VM) deployment, and the Basic and Advanced monitoring levels also apply to a single-VM deployment.

For L2VPN services, Crosswork Network Controller monitors overall health based on the subservices. For L3VPN services, monitoring occurs at the node level.

Operational limits

Crosswork Network Controller applies a rate-limiting process to manage service monitoring requests efficiently. Publishing of service monitoring requests can be delayed if the number of requests raised per minute exceeds these thresholds:

- **L2VPN services**: 50 Basic Monitoring requests per minute per service, and 5 Advanced Monitoring requests per minute per service.
- **L3VPN services**: 500 Basic Monitoring requests per minute per vpn-node, and 100 Advanced Monitoring requests per minute per vpn-node.

Rate limiting also extends to monitoring data. For example, during a restore process, when all Data Gateways send data to the Tracker component, the rate at which the Tracker processes and forwards data to Assurance Graph Manager is regulated. This can delay the reporting of Events of Significance (EOS) following the restore. An event with a severity level of warning notifies you of the delay, and the event clears when Crosswork Network Controller resumes normal publishing of monitoring requests.

When storage reaches 70% of its capacity, an alarm indicates potential depletion, and the system automatically deletes the oldest one day of data to free up space.

Heuristic Packages

Crosswork Network Controller uses a set of rules, expressed in low-code format and saved in packages called Heuristic Packages, to monitor the health of services:

- A Heuristic Package contains what to monitor, how to compute the monitored metrics, and the symptoms associated with service health degradation. The overall health of the service is determined by applying the rules from the Heuristic Package.
- The default Heuristic Packages provided with Crosswork Network Controller are referred to as system packages and cannot be altered. Crosswork Network Controller uses these system packages' predefined rules to deploy testing probes, including Y.1731, TWAMP, SR-PM, and Provider Connectivity Assurance (previously known as Accedian Skylight), to evaluate service health and determine whether the service complies with the Service Level Agreement (SLA). SLA compliance applies only to Provider Connectivity Assurance probes.
- If the default system packages do not fully meet your needs, you can customize them. Export an existing package, modify it, and import it to create a custom Heuristic Package.

Monitoring type filtering for VPN services lets you display VPN services based on their monitoring type, Basic or Advanced. This filtering enables rapid identification of the Heuristic Package applied to each service and supports efficient monitoring management. For example, to review all VPN services configured with Advanced monitoring for compliance checks, apply the Advanced filter to generate a focused list without manually reviewing individual service configurations.

Extended CLI support using Crosswork Network Controller system device packages provides more comprehensive service monitoring. These packages can derive exact sensor paths for metric health calculation and can be installed as a bundle.

To add or extend CLI-based KPI collections, you need support from Cisco Professional Services. Engage with your Cisco account team for more details.

Getting started

Describes the stages to set up a Crosswork Network Controller environment and start monitoring service health, including optional advanced use cases.

Service Health is available as part of the Crosswork Network Controller Advantage Package. Refer to the *Get Started* chapter in the Crosswork Network Controller 7.2.xInstallation guide.

Summary

You need a functional Crosswork Network Controller environment with devices onboard and services provisioned before you can start monitoring services. This workflow includes links to documents and processes needed to accomplish those tasks, which are beyond the scope of this document.

Workflow

To set up and start monitoring services, complete Steps 1 through 6. Steps 7 to 9 are optional and cover advanced use cases.

1. Install the Crosswork Network Controller Advantage package.
   - Refer to the Crosswork Network Controller 7.2.x Installation guide.
2. Do the basic reachability checks from the Crosswork Network Controller UI.
   - Refer to the Setup workflow in the Crosswork Network Controller 7.2.x Administration guide.
3. Create and provision the required Layer 2 VPN (L2VPN) and Layer 3 VPN (L3VPN) services. You can create and provision services using the Crosswork Network Controller UI or APIs.
   - Refer to the *Orchestrated Service Provisioning* chapter in the Crosswork Network Controller 7.2.x Solution Workflow Guide.
   - Refer to the Crosswork Network Controller API Documentation on Cisco DevNet.
4. Determine if you would like to configure additional external storage.

   Note

   You can configure external storage at any time.

   - If youanticipate monitoring health of many services, Cisco recommends configuring external storage after you install Service Health and before you begin monitoring the services.
5. Enable health monitoring for the provisioned services to start monitoring VPN services.
6. Establish your operational processes for responding to degraded services. Examine the impacted services and subservices health, and drill down to the root cause of the service degradation.
7. (Optional) Use SR-PM to probe and monitor links and TE policies in the network, and to measure the performance metrics of TE policies and links.
8. (Optional) Use Provider Connectivity Assurance to probe Service Health. External probes from Provider Connectivity Assurance can provide additional insights into the health of the service.

> **Note**
>
> Provider Connectivity Assurance integration is available as a limited-availability feature in this release. Engage with your account team for more information.

9. (Optional) Customize and import Heuristic Packages. Service Health offers a default set of Heuristic Packages for monitoring. If these packages do not fully meet your needs, you can customize them to align with your specific requirements.

Service health monitoring workflows

Introduces the set of Service Health workflows that guide you through monitoring scenarios, from managing stored data to customizing Heuristic Packages.

Service health monitoring workflows are guided procedures that:

- Group related tasks for common monitoring scenarios,
- Describe the stages required to complete each scenario, and
- Link to the detailed tasks and concepts that support each stage.

Available workflows

Use these workflows to perform procedures for different scenarios and functionalities:

- Managing stored data.
- Analyzing the cause of service degradation.
- Using SR-PM to monitor links and TE policies.
- Monitoring Service Health using Provider Connectivity Assurance.
- Customizing Heuristic Packages.

Managing stored data

Describes the stages to monitor storage usage and reduce the amount of monitoring data stored when Service Health storage approaches its capacity.

Crosswork Network Controller provides 50 gigabytes (GB) of storage for monitoring data. If storage reaches its limit, the system deletes the least recently used monitoring data first.

Summary

When the storage reaches 70% capacity, Crosswork Network Controller generates an alarm prompting you to configure external storage. Service Health automatically deletes the least recently used files when 80% of the 50 GB storage capacity is reached. These stages describe how to monitor storage usage. reduce the amount of data being stored, and how to add additional external storage

Workflow

These stages describe how to manage stored data:

1. Reduce the number of services being monitored by stopping the monitoring for a few services. Review the monitoring data already stored on your system, and delete any data you do not need to free up storage space.
2. Switch services that are using Advanced Monitoring to Basic Monitoring to monitor the services in less detail.
3. Configure additional external storage on Amazon Web Services (AWS) Cloud for additional storage.

Analyzing the cause of service degradation

Describes the iterative stages to examine impacted services and subservices and to determine the root cause of a service degradation.

Analyzing the cause of service degradation is an iterative, operational workflow.

Summary

Explore the impacted services and subservices health, and examine the root cause of the service degradation using any of these methods.

Workflow

These stages describe how to analyze the cause of service degradation:

1. View monitored services and identify degraded services.
2. Identify the cause of the service degradation by inspecting a degraded service using the last 24-hour metrics, determining its active symptoms, and identifying root causes using the Assurance Graph.
3. Confirm whether the reported degradation is a valid issue. If the issue is not valid, you may need to adjust the monitoring level, Basic Monitoring or Advanced Monitoring, to ensure accurate reporting of a service's health. Alternatively, modify the system Heuristic Package to create a custom Heuristic Package, which helps resolve false-positive reporting of a service's health. If the reported issue is valid, proceed to the next stage.
4. Analyze whether the service degradation is on account of an issue with device health by viewing the devices participating in the service and the collection jobs.

Using SR-PM to monitor links and TE policies

Describes the stages to enable Segment Routing Performance Measurement collection and view the performance metrics of links and TE policies.

To measure the performance metrics of links and TE policies, Crosswork Network Controller can leverage Segment Routing Performance Measurement (SR-PM).

Summary

When this feature is enabled, Crosswork Network Controller gathers and processes additional metrics such as Delay, Delay Variance, and Liveness to compute the health and determine whether any of the metrics have crossed the threshold compliance.

Workflow

These stages describe how to enable SR-PM collection and view the performance metrics collected using SR-PM:

1. Enable SR-PM metrics collection in Crosswork Network Controller.
2. View the performance metrics of the links and TE policies.
3. Ensure that the health of the policy or link is reported accurately without any false issues. If false reporting of degradation is observed, you can create a custom Heuristic Package by modifying the system Heuristic Package to provide customized and accurate health reporting.

Monitoring service health using provider connectivity assurance

Describes the stages to install Provider Connectivity Assurance, add it as a provider, set up probe sessions, and view L3VPN performance metrics.

Crosswork Network Controller can use external probing from Cisco Provider Connectivity Assurance (formerly Accedian Skylight) to measure the performance metrics of the L3VPN services.

> **Note**
>
> Monitoring L3VPN services using Provider Connectivity Assurance is supported only with Advanced monitoring and requires a Provider Connectivity Assurance Essentials license. Refer to Provider Connectivity Assurance Licensing Tiers for more information.

Summary

The performance metrics of the L3VPN services are compared with the contracted service-level agreement (SLA, defined in the Heuristic Package), with the results accessible on the UI for further analysis.

Workflow

To add Provider Connectivity Assurance as a provider in Crosswork Network Controller, follow Stages 1 and 2. Follow the remaining stages iteratively for operational purposes.

1. Install the Provider Connectivity Assurance solution.
   - Refer to the Cisco Crosswork Assurance and the Cisco Crosswork Assurance installation guide for information on installing the Provider Connectivity Assurance solution and deploying it with Crosswork Network Controller.

   > **Note**
   >
   > Sign up and create an account with the self sign-up tool to access the Provider Connectivity Assurance documentation.

2. Add Provider Connectivity Assurance as a provider in Crosswork Network Controller.
3. Set up probe sessions for the L3VPN service.
4. View the metrics in the Crosswork Network Controller UI.
5. Analyze the cause of the service degradation.
6. Confirm whether the reported degradation is valid. If it is not valid, modify the system Heuristic Package to create a custom Heuristic Package for a customized report on service health.

Customizing Heuristic Packages

Describes the stages to analyze monitoring requirements, export a system package, build and import a custom Heuristic Package, and verify the results.

Crosswork Network Controller 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. Heuristic Packages provided by the system are read-only and cannot be modified.

Summary

If you find that the Heuristic Packages provided by the system do not meet your monitoring requirements, in terms of monitoring metrics or monitoring thresholds, you can create a customized Heuristic Package that caters to your specific monitoring requirements using the procedures in this workflow.

Customizing Heuristic Packages is not included in the standard Day 2 support responsibilities. For assistance, reach out to the Cisco account team or contact Cisco Professional Services.

Workflow

These stages describe how to customize Heuristic Packages:

1. Analyze your network services to identify monitoring requirements. Check the system Heuristic Packages for rules, subservices, and metrics to ensure that the system packages do not already include the necessary metrics, services, or thresholds. Determine the package that most closely matches the conditions you want to identify in your network.
2. Export the package or packages that include the functions you want to use.
3. Using the supplied packages as your template, build a new package that gathers the data you need to make determinations about the health of the service you want to monitor. In the simplest use case, you may only need to edit the threshold points based on the service-level agreements (SLAs) used in your network. In more complicated use cases, you might need to build a Heuristic Package from scratch.
4. Import the customized Heuristic Package in Crosswork Network Controller.
5. Apply the custom package to each service that requires it.
6. Verify that the custom package is providing the monitoring data that you need to meet your requirements.

Service Health audit logging

Explains how Service Health audit logging records the source IP address of configuration changes to provide a comprehensive audit trail for accountability and security.

Service Health audit logging is an enhanced logging capability that:

- Records the source IP address for specific Service Health changes,
- Tracks the originating IP address of users who make modifications using either the UI or the API, and
- Provides a comprehensive audit trail to maintain accountability and strengthen operational security.

Actions that record the source IP

Service Health audit logs record the source IP for these actions:

- **Importing custom Heuristic Packages**: Identifies the source of new or updated custom packages.
- **Enabling or disabling service monitoring**: Records who initiated changes to the monitoring status of services.
- **Changing monitoring levels**: Tracks modifications to the granularity or type of service monitoring.

This logging capability provides a clear record of administrative actions and serves purely for auditing purposes without impacting system operation.

View the source IP in Service Health audit logs

Outlines how to access and review enhanced Service Health audit logs in the Crosswork Network Controller UI, highlighting the ability to view the source IP address for configuration changes and identifying the originating IP of the user who performed each action.

The inclusion of the source IP in Service Health audit logs does not require explicit configuration. You can access and review these enhanced audit logs through the Crosswork Network Controller UI.

Procedure

1. From the main menu, choose **Administration** > **Audit Logs**.
2. Filter or search for entries related to Service Health, Heuristic Packages, or service monitoring actions.
3. Review the audit log entries for these actions, which now include the source IP address.

   This information helps identify the originating IP of the user who performed the action.

Service Health monitoring scale information

Provides the supported monitoring scale for Service Health, including the maximum number of services for Basic and Advanced monitoring.

Understand the scale limitations and monitoring capacities of Service Health, including the maximum supported number of services for both Basic and Advanced Monitoring configurations.

You can monitor a maximum of 52,000 services in total. You can monitor either 52,000 services using only Basic Monitoring, or a combination of Basic and Advanced Monitoring up to 52,000 services in total, with no more than 2,000 services using Advanced Monitoring.

Table 2: Monitoring support

Type of monitoring	Supports
Basic monitoring	52,000 services
Advanced monitoring	2,000 services

Note

For large Layer 3 (L3) VPN deployments, Service Health supports either Basic or Advanced monitoring for up to three large VPNs, with a maximum of 4,000 VPN nodes and up to 20,000 endpoints per deployment.

Note

If you enable large VPN monitoring services while L2 and L3 services are still being discovered, health reporting can be delayed by up to nine hours because of heavy system load and concurrent processing. For best practices, workflow steps, and examples of VPN health reporting that typically completes in one hour, refer to the Enabling large VPN services in Service Health article.

Service Health monitoring single-VM scale information

Provides the supported monitoring scale for a Service Health single virtual machine deployment, including Basic, Advanced, L3VPN, and probe session limits.

Describe the monitoring scale supported by a Service Health single virtual machine deployment, including limits for Basic and Advanced Monitoring, L3VPN services, and probe sessions.

You can monitor a maximum of 2,200 services using Basic Monitoring and Advanced Monitoring, with 200 of those services using Advanced Monitoring. In addition, one L3VPN service (more than 200 nodes) and 200 probe sessions for end-to-end monitoring are available.

For more information on Service Health monitoring single virtual machine (VM) support, refer to the Crosswork Network Controller 7.2.x Administration guide.

Table 3: Single-VM scale information

Type of monitoring	Supports
Basic Monitoring	2,000 services
Advanced Monitoring	200 services
L3VPN (up to 200 nodes)	1 service
Probe sessions for end-to-end monitoring	200 sessions

3 Monitor Service Health

Topics:

- Start Service Health monitoring
- Adjust monitoring settings
- SR-PM monitoring for links and TE policies
- Service health monitoring using CS-SR policies
- Service health monitoring using Cisco provider connectivity assurance

This chapter covers the following topics:

Start Service Health monitoring

Enable health monitoring for a provisioned L2VPN or L3VPN service, and select a monitoring level and configuration profile.

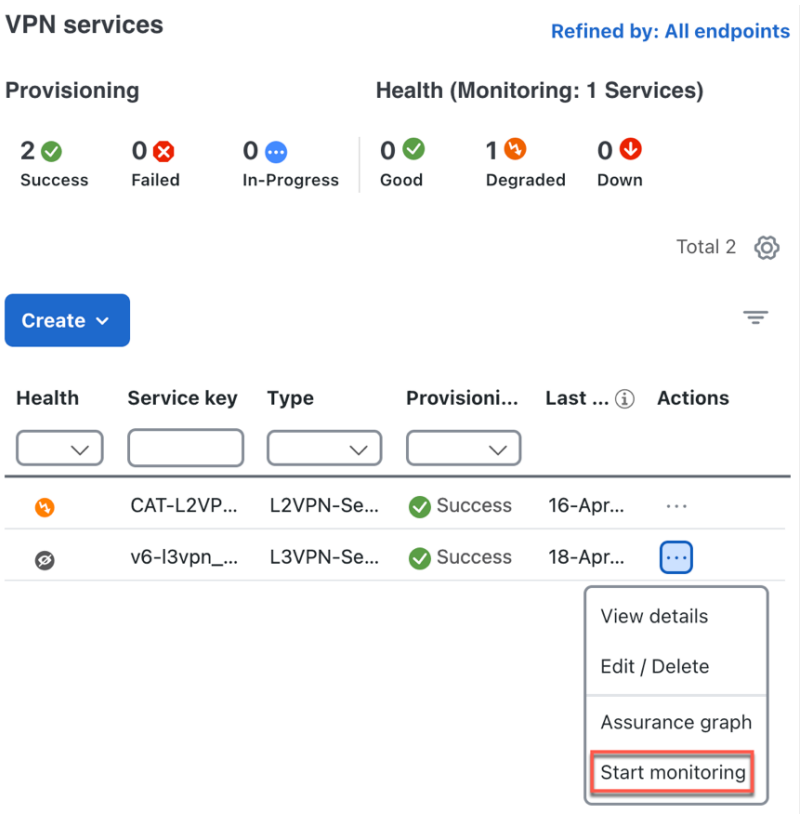
Start monitoring the health of a service.

Before you begin

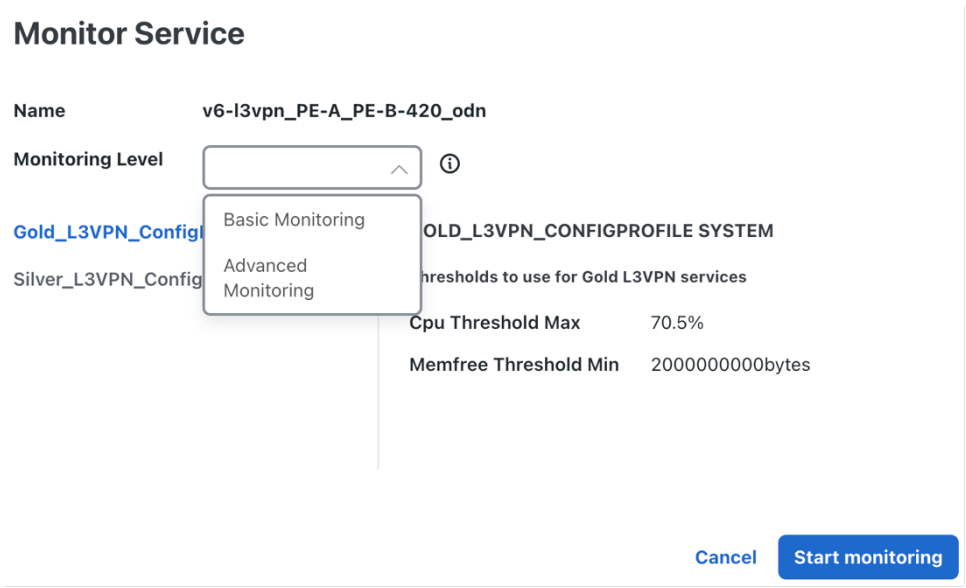
This procedure assumes that you have already provisioned L2VPN and L3VPN services. To create and provision services, refer to the Orchestrated Service Provisioning chapter in the Cisco Crosswork Network Controller 7.2.x Solution Workflow Guide.

Procedure

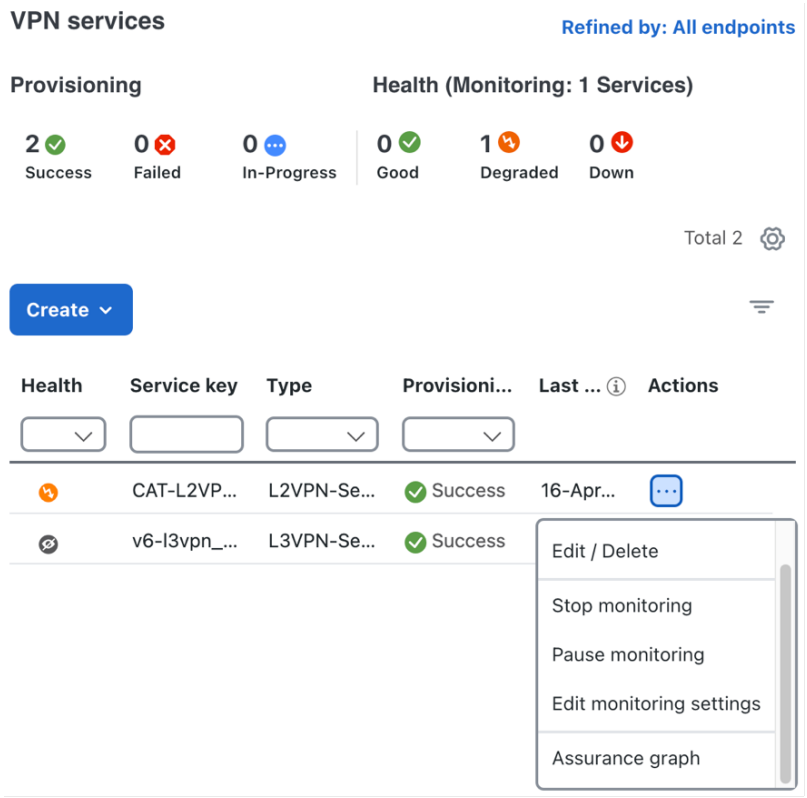
1. From the main menu, choose **Services & Traffic Engineering** > **VPN Services**. The map opens on the left side of the page and the table opens on the right side.
2. For a service that is not currently monitored, indicated by a gray icon in the **Health** column, click ⋯ in the **Actions** column.
3. Click **Start monitoring**.



4. In the Monitor Service window that appears:
   a) Select the **Monitoring level** as **Basic Monitoring** or **Advanced Monitoring**.
   b) Select a configuration profile from the list to apply it for monitoring the service.



5. Click **Start monitoring**. The **Health** column reflects the health of the service.



What to do next

If the health of the service is degraded, identify the root cause of the degradation and take measures to correct the issue.

Adjust monitoring settings

Introduces the monitoring settings that you can use to adjust Service Health monitoring for a service.

These topics explain the various monitoring settings that you can use to adjust Service Health monitoring:

- Editing the monitoring level and configuration profile,
- Pausing and resuming monitoring, and
- Stopping monitoring and choosing whether to retain historical data.

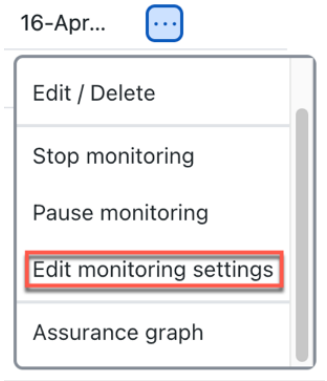
Edit the monitoring settings

Update the monitoring level or the configuration profile that Crosswork Network Controller uses to monitor a service.

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, changing the Gold profile to the Silver profile or the Silver profile to the Gold profile. Refer to About Heuristic Packages for information about configuration profiles.

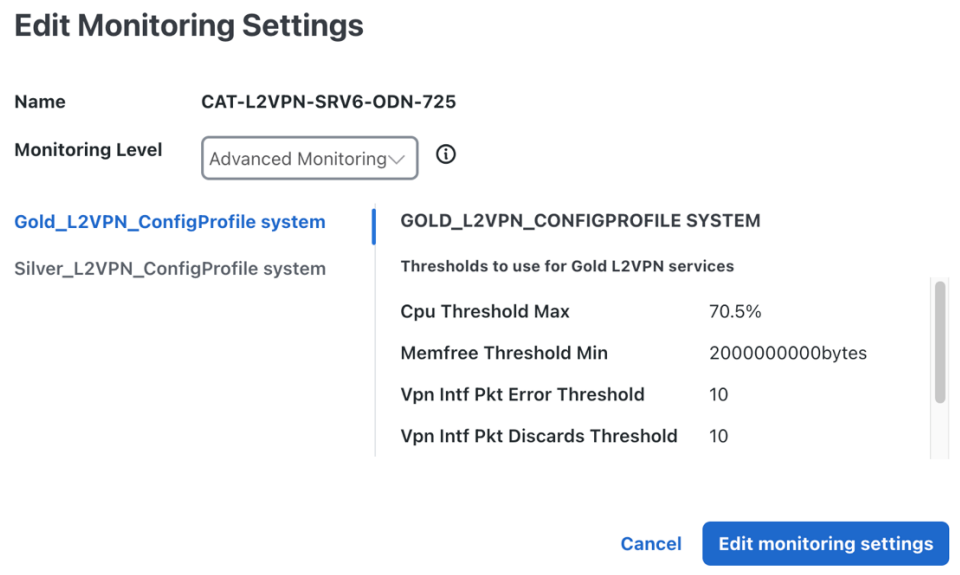
Procedure

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.
2. In the **Actions** column, click ⋯ for the service for which you want to edit the monitoring settings.
3. Choose **Edit monitoring settings** from the menu.



The Edit monitoring settings dialog box appears.

4. Choose the **Monitoring level** or the **Configuration profile**, as required.



Note

When you switch between Advanced and Basic Monitoring, it can take over 15 minutes for subservice health and active symptoms to become visible.

5. Click **Edit monitoring settings**.

   A confirmation dialog box appears.

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

   Crosswork Network Controller starts monitoring the service's health using the updated values.

What to do next

If Crosswork Network Controller reports that the health of the service is degraded, identify the cause of the service degradation and take corrective measures. Refer to Analyze Service Health for more information.

Pause and resume Service Health monitoring

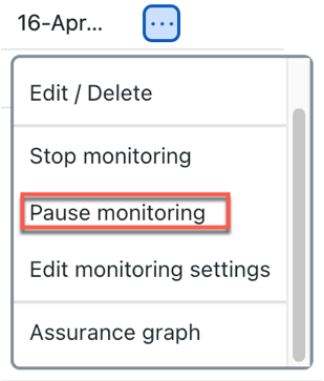
Provides instructions for pausing and resuming Service Health monitoring. Outlines when and why to temporarily stop health monitoring for a service, what data is impacted during the pause, and how to restore monitoring with existing rules and profiles.

Pause and then resume monitoring the health of a service.

With this option, you can temporarily pause monitoring the health of services. This approach is useful if a service is down because of a reported outage or scheduled maintenance and you do not want to receive notifications about the degradation. If you pause and then resume monitoring, the system continues using the same Basic or Advanced Monitoring rules and profile options as before the pause. Additionally, historical data and End of Service (EOS) are preserved in the service's history. However, because no data is collected while monitoring is paused, there are gaps in the historical data for the periods when monitoring was paused.

Procedure

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.
2. In the **Actions** column, click ⋯ for the service that you want to pause the monitoring for.
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 shows only the top-level service with a paused icon badge and no child subservices underneath.

4. In the **Actions** column, if you now click ⋯ for the service that you paused, the **Resume monitoring** option appears. Click this option to resume monitoring the health of the service.

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

   When Crosswork Network Controller resumes monitoring a service after a pause, it uses the same monitoring rules and profile options that were in place before the pause.

Stop Service Health monitoring

Provides instructions for stopping health monitoring on a service, including guidance on retaining or purging historical monitoring data and step-by-step actions for modifying monitoring settings within the service interface.

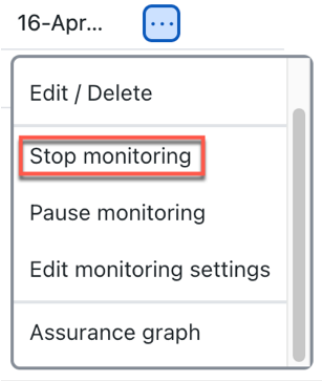
Stop monitoring the health of a service.

When you choose to stop monitoring a service, the system prompts you to confirm whether you want to retain the historical monitoring data. These options are available:

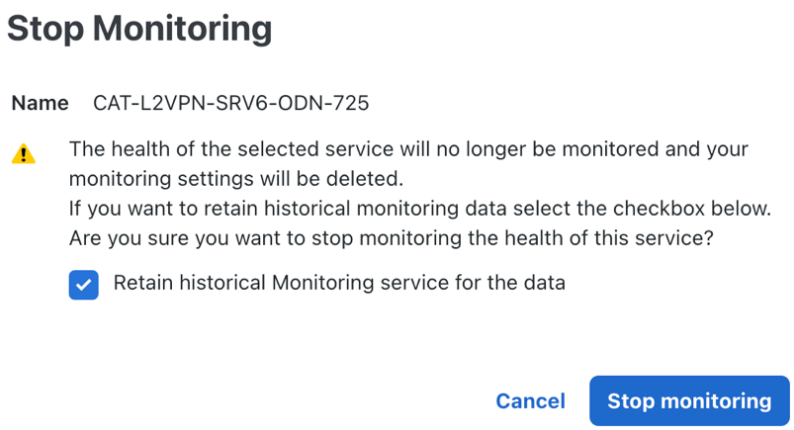
- **Retain historical data**: If you choose to retain the historical data, all monitoring information collected prior to the stoppage remains accessible. This data is preserved and available for analysis when monitoring is resumed. The monitoring settings are also retained, ensuring a seamless transition back to active monitoring with historical context.
- **Do not retain historical data**: If you decide not to retain the historical data, all monitoring settings and historical data are purged from the database. This action also deletes the Assurance Graph for the stopped service. Subsequent monitoring of the service starts anew, without any reference to previous data.

Procedure

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.
2. In the **Actions** column, click ⋯ for the service that you want to stop monitoring.
3. Choose **Stop monitoring** from the menu.



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



5. Click **Stop monitoring**.
6. If you stopped monitoring a service and selected **Retain historical monitoring service for the data**, you can later start monitoring that service with historical data still available. From the **Actions** column of the service, click ⋯ and select **Start monitoring**.

SR-PM monitoring for links and TE policies

Describes how Service Health uses the Segment Routing Performance Measurement feature to measure the performance metrics of links and Traffic Engineering policies.

To measure the performance metrics of links and Traffic Engineering (TE) policies, Segment Routing Multiprotocol Label Switching (SR-MPLS) and Resource Reservation Protocol-Traffic Engineering (RSVP-TE), Service Health uses the Segment Routing Performance Measurement (SR-PM) feature. This feature enhances the capabilities for troubleshooting and health analysis by providing detailed, historical, and consolidated views of links in the network and transport path metrics. This enables network and service operators to proactively manage, troubleshoot, and optimize the network infrastructure.

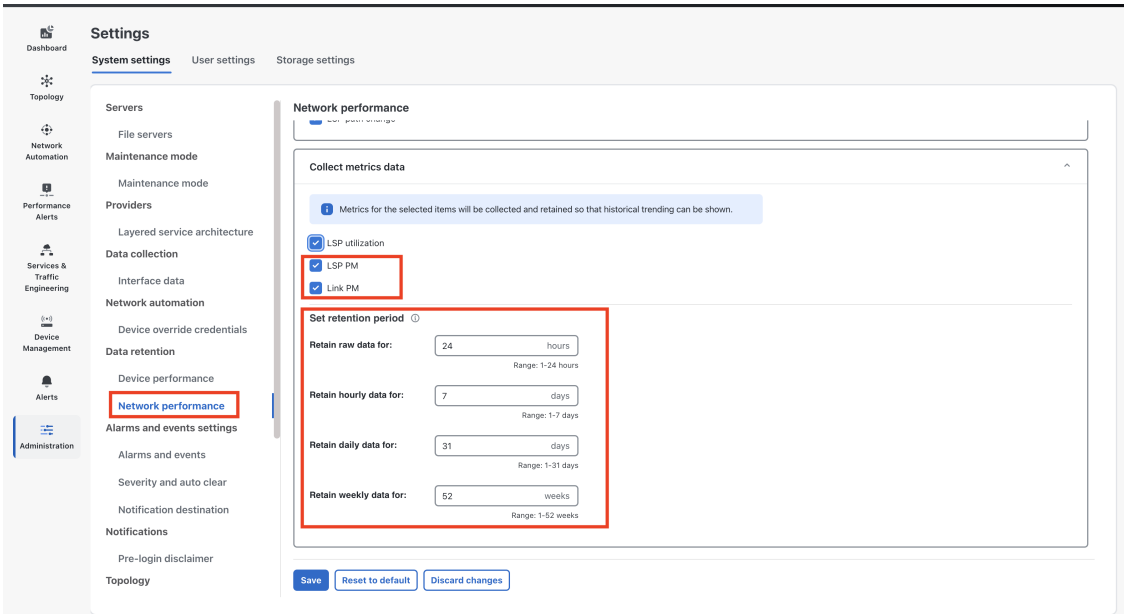
Enable SR-PM metrics collection

Provides instructions for enabling SR-PM metrics collection on SR policies and links, and outlines how to configure data retention settings to maintain historical network performance trends.

Enable the SR-PM metrics collection.

Procedure

1. From the main menu, select **Administration** > **Settings** > **Data retention** > **Network Performance**. The **Network performance** pane opens on the right.
2. Under **Collect metrics data**, select:
   - **LSP PM** to enable metrics collection for SR policies.
   - **Link PM** to enable metrics collection for links.



3. To retain historical data and view trends of these metrics, select the duration for which data should be collected and retained.

> **Note**
>
> Metric data is collected and retained only for the options for which you enabled SR-PM metric collection.

View TE policy performance metrics

Outlines how to view KPI metrics, status, and historical trends for TE policies using SR-PM data in Crosswork Network Controller, provides instructions for accessing policy performance, historical data, and interpreting health indicators.

SR-PM data collection is supported for SR-MPLS, SR-CS, and RSVP-TE policies. The metric data is used to assess policy health and to indicate whether any metrics have violated the SLAs defined in the Heuristic Package. You can view the KPI

metrics and the operational and administration status of the service on the policy tab in the **Service Details** page. If you enabled data retention, the historical data and trends are available in the **History** tab.

These metrics are collected for TE policies when SR-PM collection is enabled:

- Delay, available only for SR-MPLS and RSVP-TE policies.
- Delay Variance (jitter), available only for SR-MPLS and RSVP-TE policies.

> **Note**
>
> The RSVP policy under the **Transport** > **RSVP** tab in VPN Services and the **Traffic Engineering** > **RSVP** tab represent the same Traffic Engineering policy. Both pages display RSVP Performance Measurement (PM) metrics with identical values. However, the Threshold label appears only in the VPN Services Transport tab when Service Health monitoring is enabled and the device has delay measurement configured for the policies.

Before you begin

Ensure that you have completed these steps to view metrics from SR-PM collection:

- Added devices and TE policies, and created device groups.
- Enabled SR-PM collection in Crosswork Network Controller and, optionally, enabled data retention to view historical data and trends.
- Enabled SR-PM metric collection on devices.

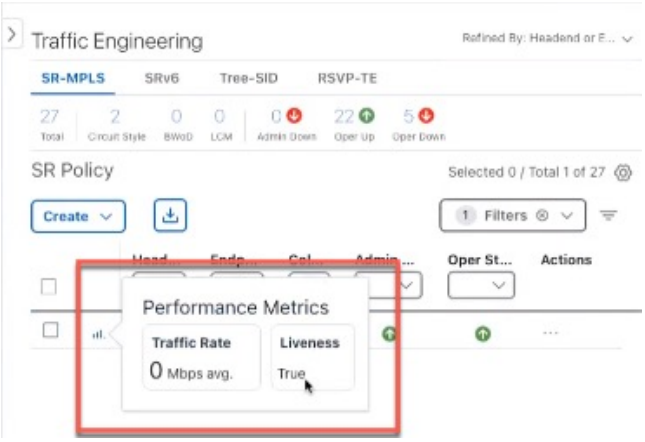
> **Note**
>
> Refer to the device-specific documentation for details. These details are beyond the scope of this guide.

Procedure

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

   Figure 1: SR-MPLS policy performance metrics in the Traffic Engineering table



3. Hover your mouse over the graph icon to view the KPI metrics in a carousel view. Alternatively, locate the policy that you are interested in from the TE table. In the **Actions** column, click ··· > **View Details**. The **Service Details** page opens and displays the KPI metrics for the policy in the **Performance metrics** section.
4. To view historical data, click the **History** tab. A chart showing the trends is displayed for each metric. Click a time frame in the chart to view the trend of the policy during the selected period.

View link performance metrics

Outlines how to view interface metrics for a network link, including bandwidth, delay, jitter, and packet loss, and provides instructions for analyzing historical trends through the topology interface in the device management system.

Link interface metrics are a set of indicators that measure the performance and quality of the communication between two or more network devices. They include parameters such as bandwidth, delay, jitter, and packet loss. Link interface metrics can help network administrators monitor and troubleshoot network issues, optimize network resources, and plan for future network expansion or upgrade.

Before you begin

Ensure that you have onboarded devices and created the required device groups.

Procedure

1. From the main menu, choose **Topology**.
2. Select a link to view its details in any of these ways:
   a) By clicking a link on the topology map.
   b) By clicking a link from the **Links** tab in the topology map.
   c) By clicking a link from the **Links** tab in the **Device Details** page.

   The **History** tab provides useful insights into the performance and trends of the network. You can select the time interval to analyze the data.

Service health monitoring using CS-SR policies

Describes how Crosswork Network Controller monitors the health of L2VPN point-to-point services that use Circuit-Style Segment Routing policies.

Crosswork Network Controller supports monitoring the health of L2VPN point-to-point services (only IETF:L2VPN:EVPN VPWS) using Circuit-Style Segment Routing (CS-SR) policies.

When the L2VPN service is configured to use circuit-style transport, Crosswork Network Controller automatically initiates monitoring of the service in both directions (A-Z and Z-A) using the `subservice.cssr.policy.health` subservice.

The subservice monitors and reports the **Admin Status**, **Operational Status**, and any flip-flops in the operational status. The operational state of the CS-SR policy is measured using the Liveness metric. This metric measures whether the path is live and capable of carrying traffic, providing a simpler yet effective way to ensure the health of the path.

Service health monitoring using Cisco provider connectivity assurance

Describes how Crosswork Network Controller uses external probing from Cisco Provider Connectivity Assurance to measure and analyze L3VPN service metrics.

Crosswork Network Controller can use external probing, provided by Cisco Provider Connectivity Assurance (formerly Accedian Skylight), to measure the metrics of the L3VPN services in the network. The metrics are compared with the contracted service-level agreement (SLA) (defined in the Heuristic Package), and the results are made available on the user interface (UI) for further analysis.

Monitoring L3VPN services using Provider Connectivity Assurance is only possible with Advanced Monitoring and requires a Provider Connectivity Assurance Essentials license. Refer to Provider Connectivity Assurance Licensing Tiers for more information. Sign up and create an account using the self sign-up tool to access the Cisco Crosswork Assurance documentation.

High-level flow

Provider Connectivity Assurance measures L3VPN service metrics through these high-level flow:

1. When you provision an L3VPN service with probe intent and enable service monitoring, the Orchestrator component of Provider Connectivity Assurance learns the probe intent and the probe topology from the provisioned service.

   These probe intents are supported:

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

2. Probe sessions with Provider Connectivity Assurance are set up automatically to monitor the service by invoking the relevant RESTConf APIs. The RESTConf APIs that are invoked to provision probe sessions include endpoint, session, service, and session activation. The maximum number of probe sessions per service is capped at 200 (for all connection types).
3. The gateway component of Provider Connectivity Assurance streams the probe metrics to Data Gateway, which collects this data using a parameterized collection job for Service Health over gNMI. These probe metrics are collected:

   - Forward and Reverse Delay.
   - Forward and Reverse Variance.
   - Forward and Reverse Packet Loss.

4. Metrics collected during probe sessions are analyzed. The system raises symptoms as needed and displays them on the Crosswork Network Controller UI.

Add Provider Connectivity Assurance as a provider

Sets up the credential and certificate profiles and onboards Cisco Provider Connectivity Assurance as a provider in Crosswork Network Controller.

Set up the necessary credential and certificate profiles for Provider Connectivity Assurance to facilitate secure communication using the HTTPS and gNMI protocols.

Before you begin

Ensure that you have taken care of these prerequisites before onboarding Provider Connectivity Assurance as a provider:

- Install the Provider Connectivity Assurance software. Refer to the Cisco Crosswork Assurance for information about installing Provider Connectivity Assurance and deploying it with Crosswork Network Controller.

  > **Note**
  >
  > You need an account with Provider Connectivity Assurance to access the documentation. Sign up and create an account with the self sign-up tool.

- Have these certificates from Provider Connectivity Assurance downloaded on your local system or in a folder that can be accessed by Crosswork Network Controller:
  - CA certificate
  - Client certificate
  - Client key

Procedure

1. Create a credential profile.
   a) Navigate to **Administration** > **Device Management** > **Credential Profiles** and click the Add (+) button to create a new profile.
   b) Enter a name and add these credential protocols: **HTTPS** and **gNMI**. Enter the username and password for both connections.
   c) Click **Save**.
2. Create a certificate profile.
   a) Navigate to **Administration** > **Certificate Management** and click the Add (+) button.
   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**.
3. Add Provider Connectivity Assurance 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 Provider Connectivity Assurance.
      - **Family**: Select ACCEDIAN_PROXY.
      - **Certificate profile**: Select the Provider Connectivity Assurance certificate profile.

> **Note**
>
> This field is displayed after you select the **Family** as ACCEDIAN_PROXY.

- **Connection types**: Supported protocols are automatically updated from the Provider Connectivity Assurance credential profile.
- **IP addresses**: Enter the IP address or the Fully Qualified Domain Name (FQDN).

  Important

  If the server certificates present in Provider Connectivity Assurance are generated using a Fully Qualified Domain Name (FQDN), enter the FQDN only in this field. Do not enter an IP address. Entering an IP address when the server certificates are generated with an FQDN causes issues in Provider Connectivity Assurance authentication and reachability.

- **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**.

What to do next

Confirm that the Provider Connectivity Assurance provider is reachable from Crosswork Network Controller.

Check Provider Connectivity Assurance reachability

Outlines how to verify Provider Connectivity Assurance reachability from Crosswork Network Controller and guides users through resolving certificate and HTTPS credential issues to ensure proper connection and onboarding.

Resolve all certificate and HTTPS credential issues to ensure that Provider Connectivity Assurance is reachable.

Procedure

---

1. From the main menu, navigate to **Administration** > **Manage Provider Access**.
2. Ensure that Provider Connectivity Assurance shows a green reachability status without errors.

   If there are certificate errors, the provider is displayed as **Degraded** and is not reachable.
3. Provider Connectivity Assurance might still be displayed as reachable on the Crosswork Network Controller providers list page despite these issues:

- Invalid HTTPS credentials.
- Incorrect ports, IP addresses, or credentials for the gNMI protocol (because reachability checks for gNMI are not performed).

4. After resolving any certificate or HTTPS credential issues, delete and onboard Provider Connectivity Assurance in Crosswork Network Controller again.

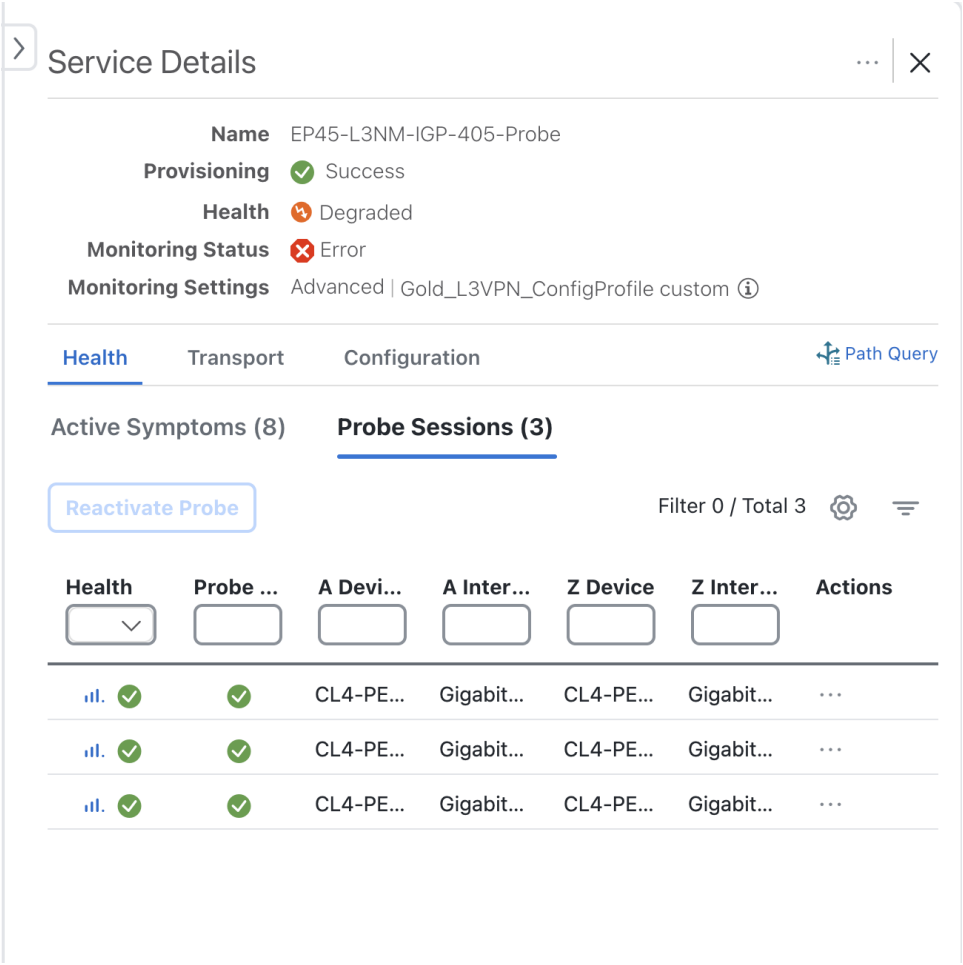
View probe session details

Outlines how to view performance metrics for Provider Connectivity Assurance and Y1731 probe sessions in L3VPN and L2VPN services, access detailed probe session data in the Probe sessions tab, and troubleshoot service health or probe session errors using the monitoring interface.

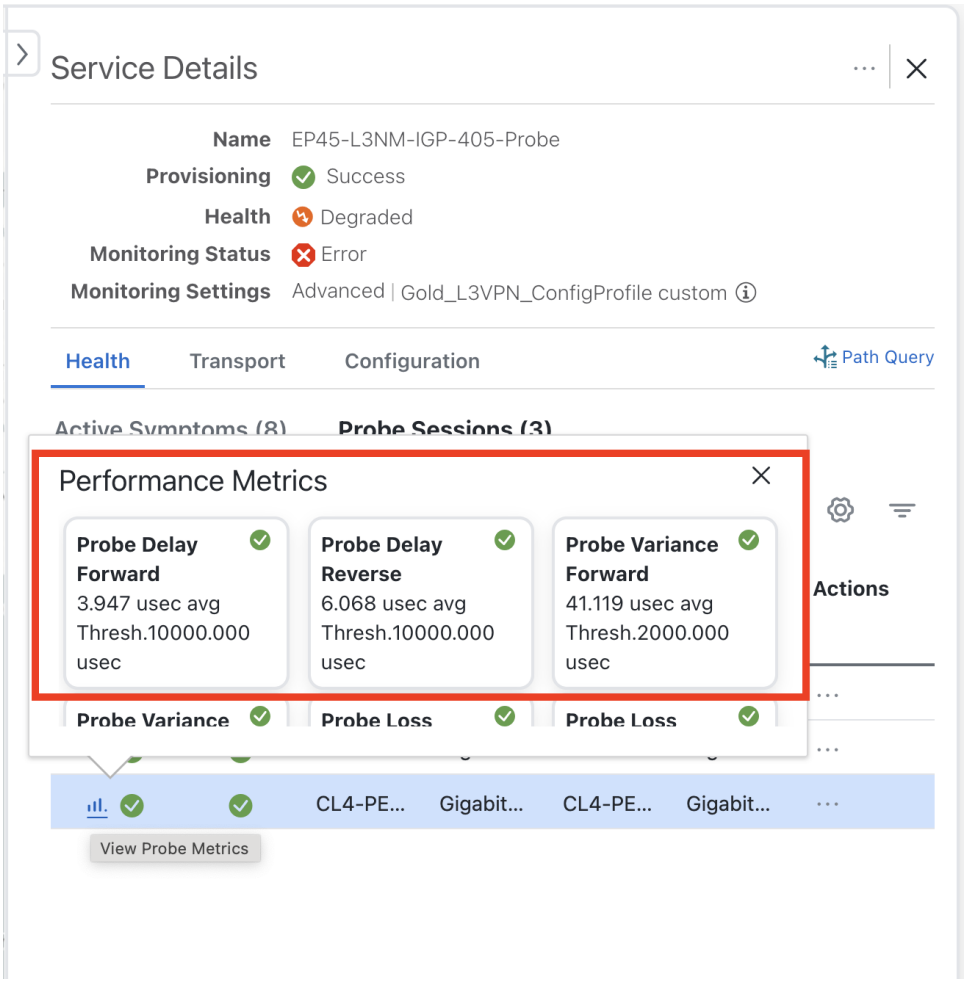
View details from Provider Connectivity Assurance probe sessions for L3VPN services and Y1731 probe sessions for L2VPN services. These details are displayed separately in the **Probe sessions** tab of the service.

Procedure

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.
2. For the service you are interested in, in the **Actions** column, click **View details**.
3. In the **Service details** page that is displayed, click the **Probe sessions** tab.



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



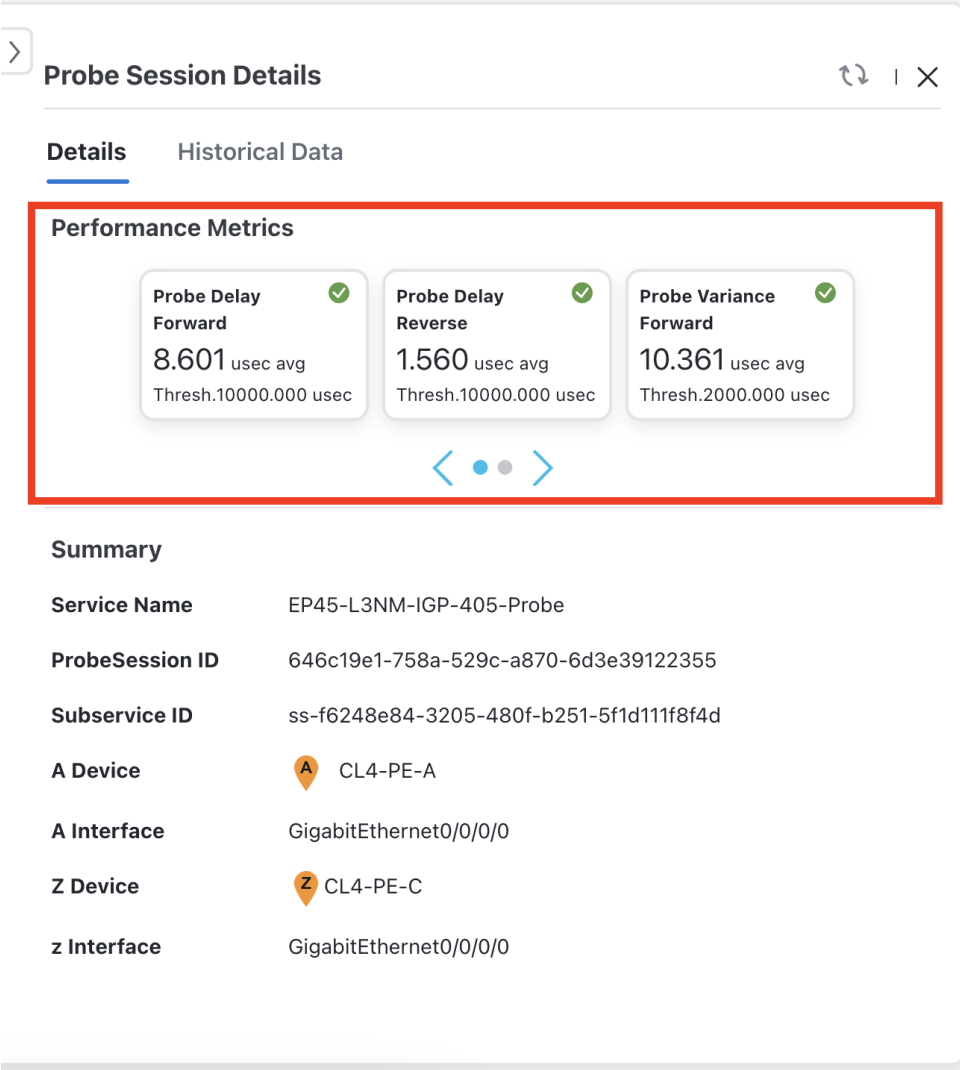
If a metric has crossed the defined threshold, a red icon is displayed in the corresponding performance metrics dashlet.

5. To view the performance metrics for a service in a carousel view, click the ⋯ icon in the **Actions** column.

The **Probe session details** window opens, displaying the metrics in a carousel view.

Note

If there are any probe provisioning errors, the monitoring status of the service is **Monitoring error**. Click **Reactivate probe** to restart the probe session for the service. If the probe session reactivates successfully, the **Probe sessions** page automatically updates with the new metrics.



The **History data** tab provides probe metrics data ranging from the past 90 days up to the most recent 24 hours. Refer to Historical data from probe sessions on page 34 for more information.

What to do next

If you find that a service is degraded, analyze the root cause of the degradation to troubleshoot the health of the degraded service. Refer to Analyze Service Health for more information.

Historical data from probe sessions

Describes how to view historical probe session data in the History data tab and the intervals used to aggregate the metrics.

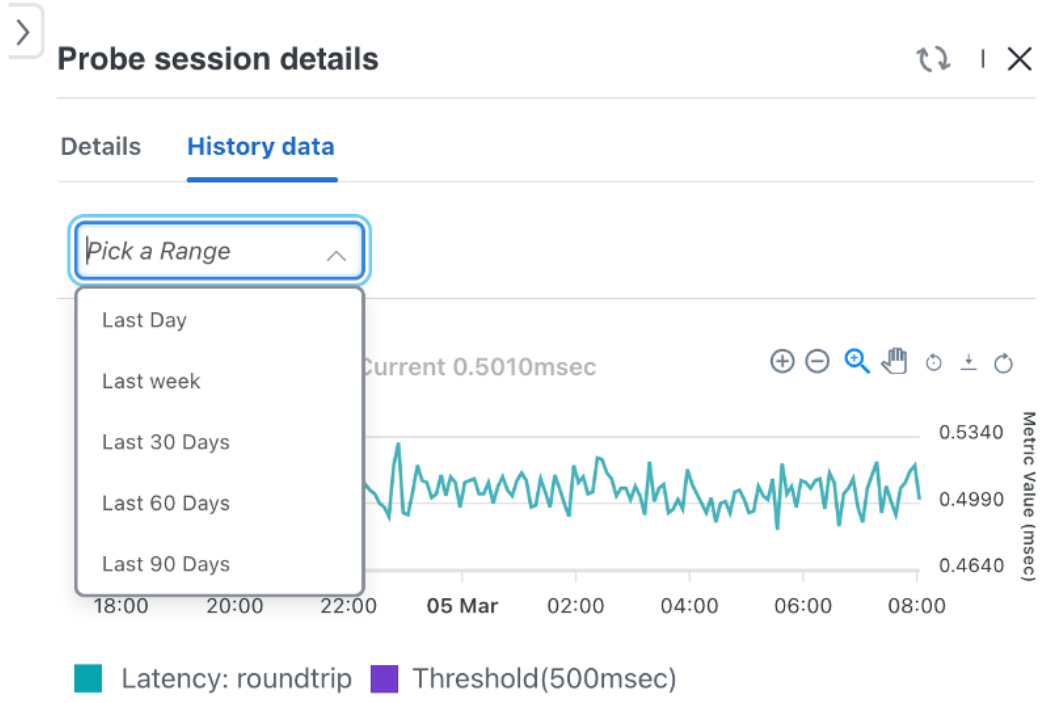
You can view the historical data from a probe session by clicking the **History data** tab in the **Probe session details** page. This tab displays data from the time monitoring was enabled for the service.

If monitoring was stopped and started again later, the **History data** tab displays data only from the time monitoring was restarted. If you chose to retain historical data while monitoring was stopped, that data is preserved and appears in the **Show History** tab of the Assurance Graph of the service, and not in this tab.

Historical probe metrics intervals

Charts displaying aggregated average metric data along with their timestamps are shown. To view data from a specific range, select the desired range from the drop-down menu.

Figure 2: Historical probe metrics chart



Note

There might be a difference in the first timestamp displayed in the historical chart metric data. This is because the timestamp displayed is in the operational time zone, while the aggregated timestamp is in UTC format with 00 hours.

This table lists the interval and aggregate interval used for each historical probe metrics period:

Table 4: Historical probe metrics intervals

Historical probe metrics period	Interval	Aggregate interval
one week (seven days)	two hours	two hours
one month (thirty days)	one day	one day (00 hours, start of the day in UTC)

Historical probe metrics period	Interval	Aggregate interval
two months (sixty days)	three days	one day (00 hours, start of the day in UTC)
three months (ninety days)	one week (seven days)	one day (00 hours, start of the day in UTC)

Known issues and limitations with Provider Connectivity Assurance

Lists the known issues and limitations when Cisco Provider Connectivity Assurance is deployed to probe the health of a service.

Describe the known issues and limitations encountered when deploying Cisco Provider Connectivity Assurance to monitor the health of a service, enabling users to understand potential challenges and take appropriate actions.

This is a list of known issues and limitations when Provider Connectivity Assurance is deployed for probing the health of a service:

1. Entering an IP address instead of the FQDN when server certificates are generated with an FQDN causes issues with provider authentication and reachability. In this case, Provider Connectivity Assurance is shown as **Degraded** in the Crosswork Network Controller providers list page (**Administration** > **Manage Provider Access**).
2. Provider Connectivity Assurance is always shown as reachable in the Crosswork Network Controller providers list page despite these issues in the Provider Connectivity Assurance credentials:
   - Invalid HTTPS credentials.
   - Incorrect ports, IP addresses, or credentials for gNMI, because there are no reachability checks for gNMI.

   In these cases, services monitored by Provider Connectivity Assurance probes have the health as **Degraded** with the symptom as *Provider Connectivity Assurance provider does not exist in DLM*. The symptoms are not cleared until you add Provider Connectivity Assurance again, and pause and resume the service monitoring.
3. When monitoring is enabled for a service with probe intent but Provider Connectivity Assurance 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.
4. You cannot delete Provider Connectivity Assurance when a probe session is active.
5. 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 Analyze Service Health

Topics:

- Monitored services overview
- VPN Services page
- Service Health Dashboard
- Monitoring status of services
- Identify the active symptoms and root causes of a degraded service
- Assurance graphs
- View the devices participating in the service
- View the collection jobs

Use Service Health monitoring to analyze service health with the metrics shown in the UI. Investigate degraded services and subservices to identify the root cause of service degradation.

Monitored services overview

Outlines the overview and navigation of monitored VPN service statuses in Crosswork Network Controller, including the VPN service health dashlet, service health dashboard, and guidance for accessing filtered service views from the home page.

The **VPN service health** dashlet on the Crosswork Network Controller home page delivers an at-a-glance summary of VPN service statuses. This enables enhanced visibility and filtering for monitoring status, such as errors, in Service Health.

The **VPN Services** page presents detailed health and monitoring data for all VPN services, enhanced by customizable columns and powerful filters that facilitate focused troubleshooting.

The **Service Health Dashboard** complements these views by aggregating service metrics and highlighting SLA breaches, enabling efficient oversight of network performance and service quality.

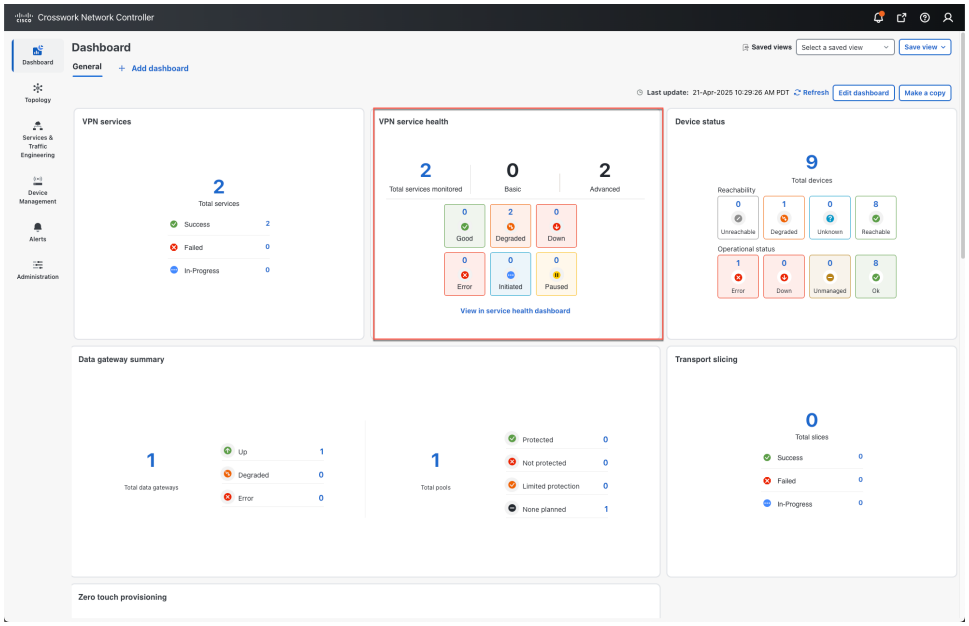
View monitored services from the Crosswork Network Controller home page

The **VPN service health** dashlet on the Crosswork Network Controller home page provides an overview of all monitored VPN services.

Click any status indicator to go to the **VPN Services** page, which displays the selected information with the applied filter.

For example, to isolate VPN services based on their monitoring configuration directly from the home page, click the number above **Advanced** services in the dashlet. The **VPN Services** page opens, filtered by the selected monitoring type.

Figure 3: VPN Service Health dashlet



> **Note**
>
> Only **Basic** or **Advanced** VPN services are shown on the VPN Services page.

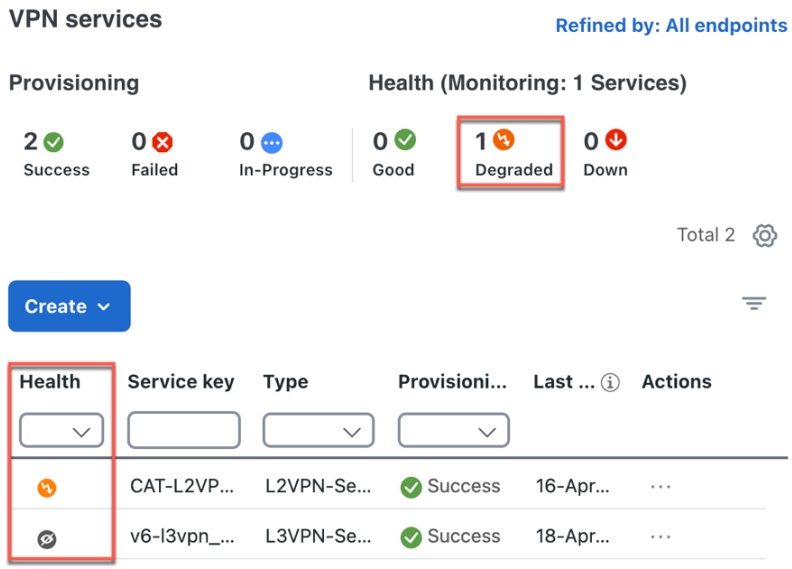
VPN Services page

Outlines how to use the VPN Services page to monitor service health, filter and troubleshoot L2VPN and L3VPN services, and customize columns, including handling monitoring errors and applying in-column filters for efficient service management.

Use the VPN Services page to monitor service health and status, and to filter and troubleshoot services experiencing degradation or monitoring errors.

The VPN Services page, located under **Services & Traffic Engineering** > **VPN Services**, provides a comprehensive overview of all VPN services, including both L2VPN and L3VPN types. The VPN services table provides direct indication of monitoring errors, allowing you to quickly identify services experiencing issues. Services with active monitoring errors are clearly marked in the table.

Degraded services show an orange icon in the **Health** column.



To clear a filter, click **X** next to the designated filter at the top of the column. To remove all filters and show all VPN services, click the **X** icon in the Filters field. Use these filters to efficiently monitor VPN service health and prioritize troubleshooting, all directly from the VPN Services page.

> **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**. Refer to Start Service Health monitoring for more information.

The VPN Services page includes these key filtering and display options:

- **Show monitoring error** check box: Quickly filter the list to display only VPN services currently experiencing monitoring errors.

  For example, if you need to focus on services with active issues, selecting this checkbox immediately narrows the view to those requiring attention.

- **Gear icon** for column visibility: Customize which columns appear in the VPN services table. Available columns include: Health, Service key, Type, Monitoring type, Provisioning state, Monitoring state, Last updated time, Date created.

  Adjusting visible columns helps you focus on the most relevant service attributes. For instance, hiding less critical columns like Date Created can declutter the interface and emphasize health and monitoring details.

- In-column filters: Refine the displayed services using these filter options:
  - **Health**: Down, Degraded, Good, Paused, Initiated, Error, Unmonitored
  - **Type**: L2VPN-Service, L3VPN-Service
  - **Monitoring type**: Basic, Advanced
  - **Provisioning state**: Success, Failed, In-Progress
  - **Monitoring status**: Error, Success

These filters allow you to isolate specific service conditions. For example, filtering Health to show only Degraded services helps prioritize troubleshooting efforts.

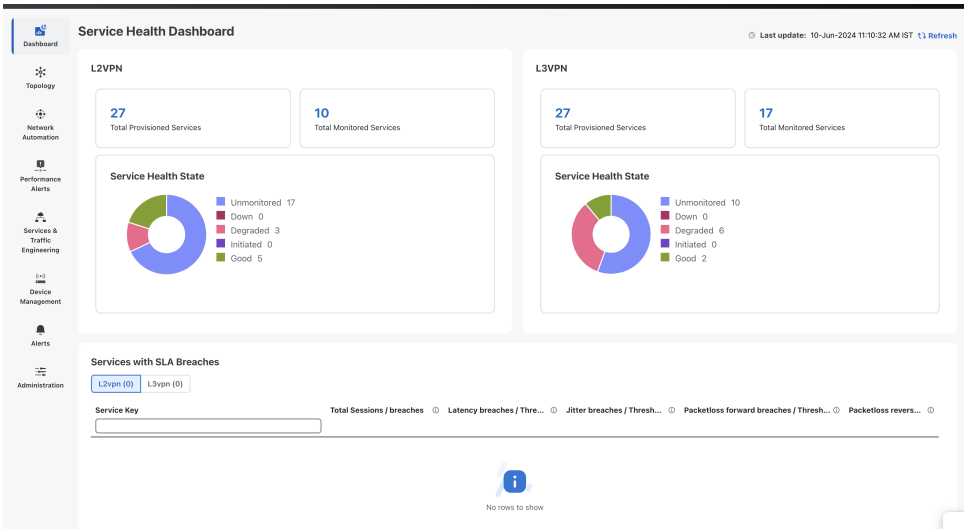
Service Health Dashboard

Introduces the Service Health Dashboard, which displays consolidated L2VPN and L3VPN metrics, active monitoring sessions, and SLA breach indicators for latency, jitter, and packet loss. Explains dashboard navigation and filtering using status indicators and the wheel graph.

To access the Service Health Dashboard from the Crosswork Network Controller home page, click **View in service health dashboard** in the **VPN service health** dashlet.

The **Service Health Dashboard** displays a consolidated view of L2VPN and L3VPN services, including the total number of provisioned services and the number of monitored services. The dashboard also displays active monitoring sessions for L2VPN and L3VPN services. It indicates SLA breaches for measured metrics such as latency, jitter, and packet loss in both directions.

Figure 4: Service Health Dashboard

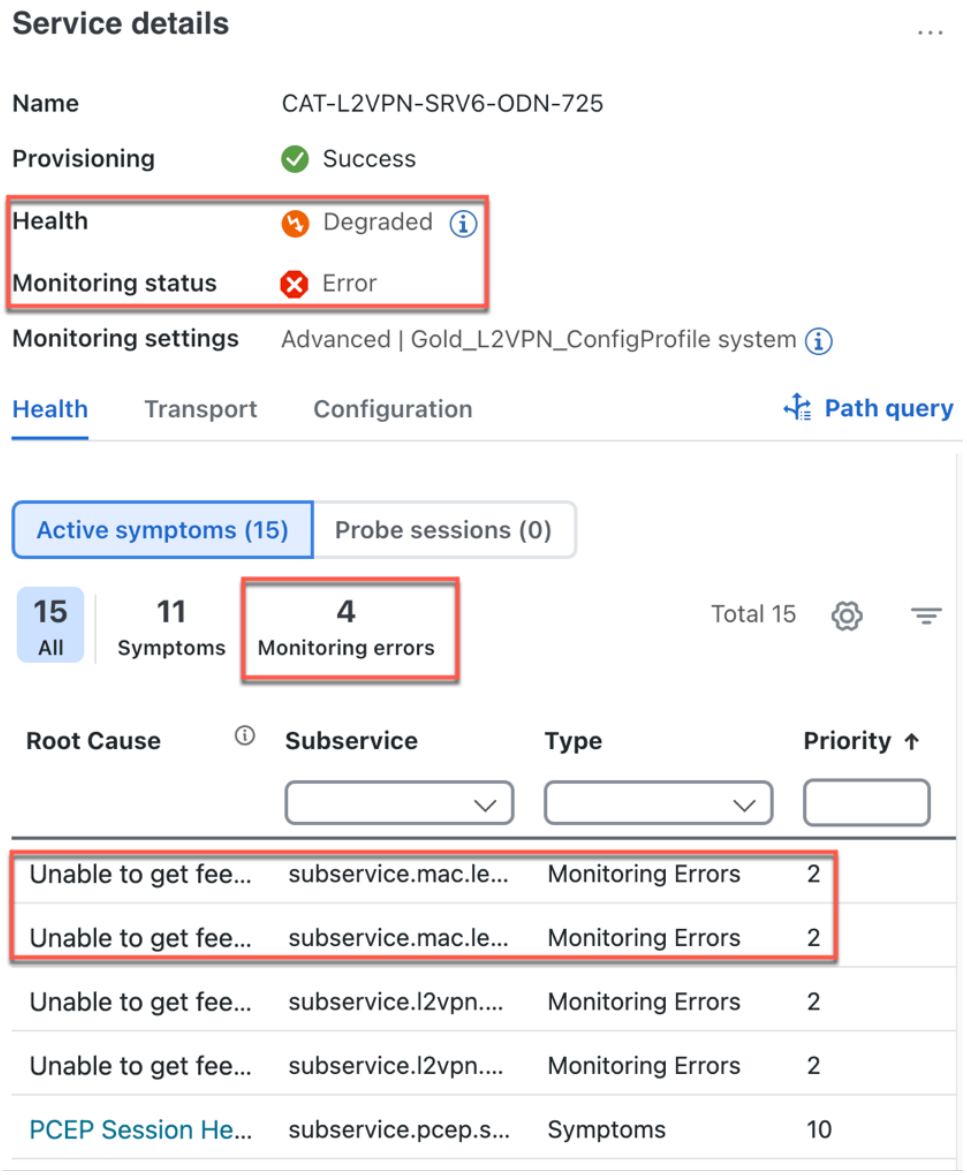


Clicking any status indicator, or different colors in the wheel graph, redirects you to the **VPN Services** page with a filter set for the selected status.

Monitoring status of services

Introduces how to monitor the status of services, including viewing service health indicators, interpreting Healthy or Error states, and understanding monitoring errors and their representation in the Assurance Graph historical timeline.

View the monitoring status of a service from its service details page. From the main menu, choose **Services & Traffic Engineering** > **VPN Services**. Locate the required service. Under the **Actions** column, click **View details**. This page displays the monitoring status and health status of the service.



A monitoring status can be **Healthy** or **Error**:

- **Healthy**: The end-to-end flow of monitoring is working as expected, and Crosswork Network Controller can evaluate service health.
- **Error**: Crosswork Network Controller cannot monitor the current health of the service due to component failures, operational errors, or device errors. The displayed health status is the last known health of the service. Filter monitoring errors using the mini dashboard or filters.

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

Monitoring errors in the Assurance Graph historical timeline

In the Assurance Graph historical timeline, EOS is displayed for monitoring errors as well. If the service is healthy and monitoring errors exist, a green warning icon appears. If the service is degraded and monitoring errors exist, an orange warning icon appears. When you click these icons, the symptoms table displays details with type set to **Monitoring errors**.

> **Note**
>
> The historical timeline displays monitoring errors only when the monitoring errors setting is enabled via API. There is no option to enable this setting from the UI. Once this setting is enabled, the system starts to log these monitoring errors as EOS and display them in the historical timeline. Refer to the API documentation on Cisco DevNet for more information.

Identify the active symptoms and root causes of a degraded service

Analyze the root cause of reported active symptoms and impacted services to determine which issues must be addressed first to maintain a healthy network setup.

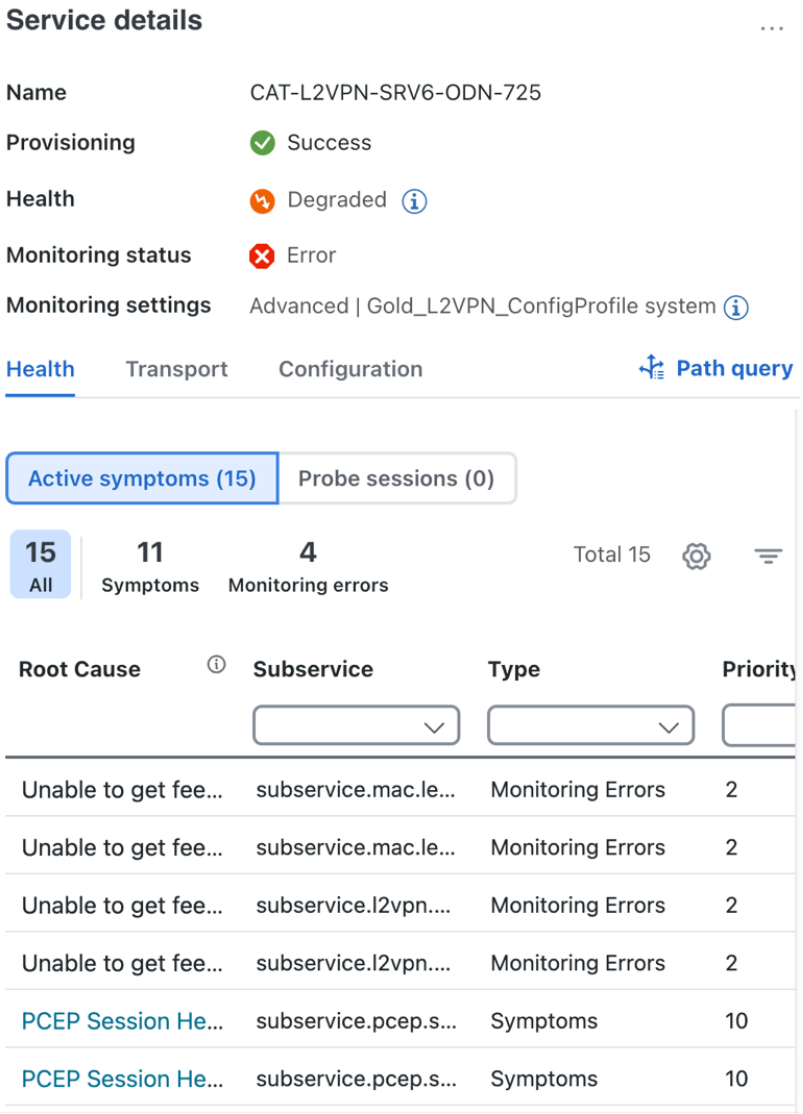
Analyze the root cause of reported active symptoms and impacted services to determine which issues must be addressed first to maintain a healthy setup and which require further inspection and troubleshooting.

> **Note**
>
> L3VPN service monitoring is supported on Cisco IOS XR devices, but not on Cisco IOS XE devices. If a provider and its devices are deleted and then added again for an L3VPN service being monitored, the service remains in the degraded state and displays a monitoring status of Monitoring error. To recover from this error, stop and restart the service monitoring.

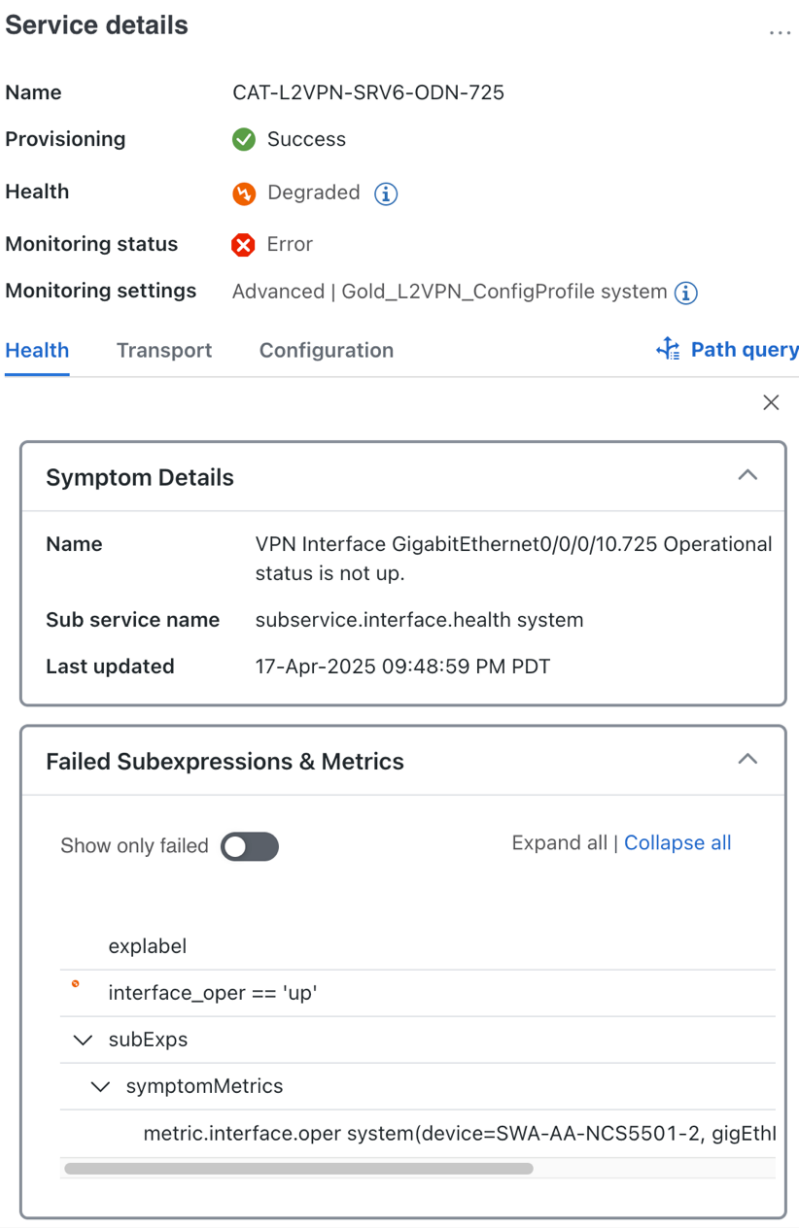
Procedure

1. From the main menu, choose **Services & Traffic Engineering** > **VPN Services**. The VPN Services page opens.
2. In the **Actions** column for the list of services, click ⋯ and select **View details**. The Service details panel appears on the right side.
3. Select the **Health** tab and then click the **Active symptoms** tab. By default, the Active Symptoms table displays **Active symptoms** and **Monitoring errors**. To display only Active Symptoms, either click the **Symptoms** tab in the mini dashboard or select **Symptoms** from the filter box under **Type**. The table will then show only Active Symptoms.

Review the active symptoms for the degraded service, which includes the Root Cause, Subservice, Type, Priority, and Last Updated details.



4. Click 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. Use the **Show only failed** toggle to display only the failed expression values.



5. Click the **Transport** and the **Configuration** tab, then review the details provided.
6. Click **X** in the top-right corner to return to the VPN Services list.

What to do next

- To monitor VPN services using Assurance Graph capabilities and inspect degraded services or related nodes, refer to Use the Assurance Graph to identify root causes.
- To identify issues with degraded services within a specific time range, refer to Identify root causes using last 24 Hr metrics.
- To identify a Service Health issue by examining the collection jobs, refer to View the collection jobs.

Assurance graphs

Introduces the Assurance Graph that visually represents a service instance and its dependent subservices, enabling you to locate problem areas and troubleshoot service degradation for both L2VPN and L3VPN services.

The Service Health status eliminates potential confusion by distinguishing between Service Health and Event Health node status. This screen applies to both L2 and L3VPNs.

In Crosswork Network Controller, a service instance comprises various subservices, each assured independently. The overall health of the service depends on the health of these subservices.

The Assurance Graph visually represents the service instances and their dependent subservices in a graph format. The topmost node in this logical dependency tree represents the monitored service instance. The child nodes represent its subservices. These subservices may further depend on other subservices.

This graphical representation helps locate problem areas and provides indications of possible symptoms and impacting metrics. It aids in troubleshooting degradation issues. Crosswork Network Controller 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**. The Assurance Graph displays the graph in the left pane and shows details of the service in the right pane. Toggle **Show History** to view the historical timeline. Each dot on the historical timeline represents one EOS for a service.

For each EOS, you can view the Assurance Graph and symptoms with 24 hours of metrics collected based on the EOS time. Forexample,ifmonitoring wasstopped for a service, a dot appears indicating that monitoring was stopped.

Clicking and dragging over a selected range on the EOS allows you to zoom in on a range of time. If you hover your mouse over a single event, a pop-up appears showing the service monitoring status for Time, Event Type, Node Name, Service Health, Symptoms, and Event Health.

The Service Health status eliminates potential confusion by distinguishing between Service Health and Event Health node status. This view applies to both L2 and L3VPNs.

Assurance Graph for L3VPN services

For L3VPN services, Crosswork Network Controller monitors the service and builds the Assurance Graph historical timeline at the node level. The historical timeline includes a summary node for each device and feature-level nodes under each summary node. Nodes with dependencies spanning other nodes, such as `path.sla.summary`, have a feature-level summary node in the historical timeline.

Select endpoints

The Assurance Graph builds its view based on the data-sending endpoints (headends) of VPN nodes. If there are more than 50 nodes in the historical timeline, Crosswork Network Controller indicates that it is too large to display. Use the **Select endpoints** option in the historical timeline toolbar to view up to 50 endpoints at a time.

The Assurance Graph filtering is based only on the VPN nodes and does not support filtering by a combination of VPN nodes and endpoints. Deselecting one endpoint will not update the historical timeline. The historical timeline updates only when both endpoints for a VPN node are removed, leading to the entire VPN node being removed from the Assurance Graph.

Note

For a service, the Transport tab displays all discovered transports related to selected VPN nodes, considering both headend and tailend roles. When you use the **Select endpoints** option and deselect a headend endpoint, the **Transport** tab updates to remove the headend endpoint from view but may still show the tailend endpoint if it is relevant to other headends.

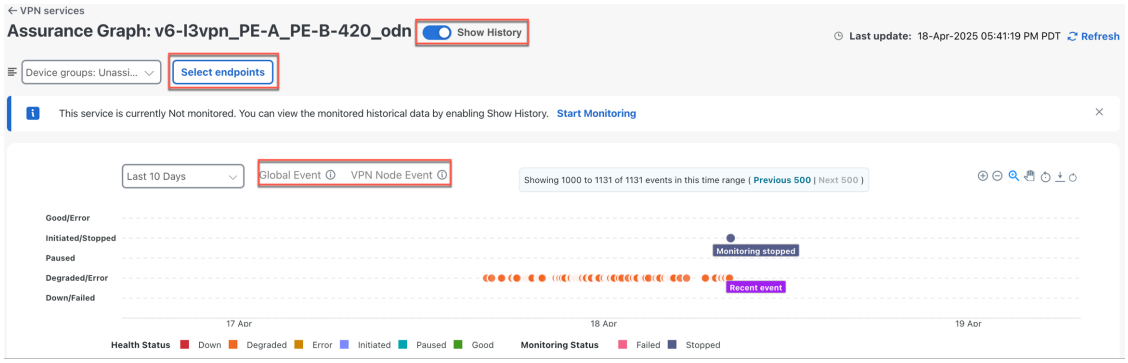
In contrast, the **Assurance Graph** focuses only on headend endpoints. If you deselect an endpoint and no other endpoints are left for that node, the graph removes the entire node from the display.

Show history

When you toggle **Show History**, you will see two types of events: **VPN node events** and **Global events**. When you hover over the EOS, the event type appears in its description.

- **Global events**: These events span multiple VPN nodes. For example, an EOS in the probe service (`path.sla.summary`) is classified as a Global event.
- **VPN node events**: These events are specific to a single VPN node.

In L3VPN services, symptom counts are shown at the VPN node level, with the Device ID (VPN node name) displayed alongside the symptoms. The timeline series in the Show History view displays these symptoms at the VPN node level (endpoint).



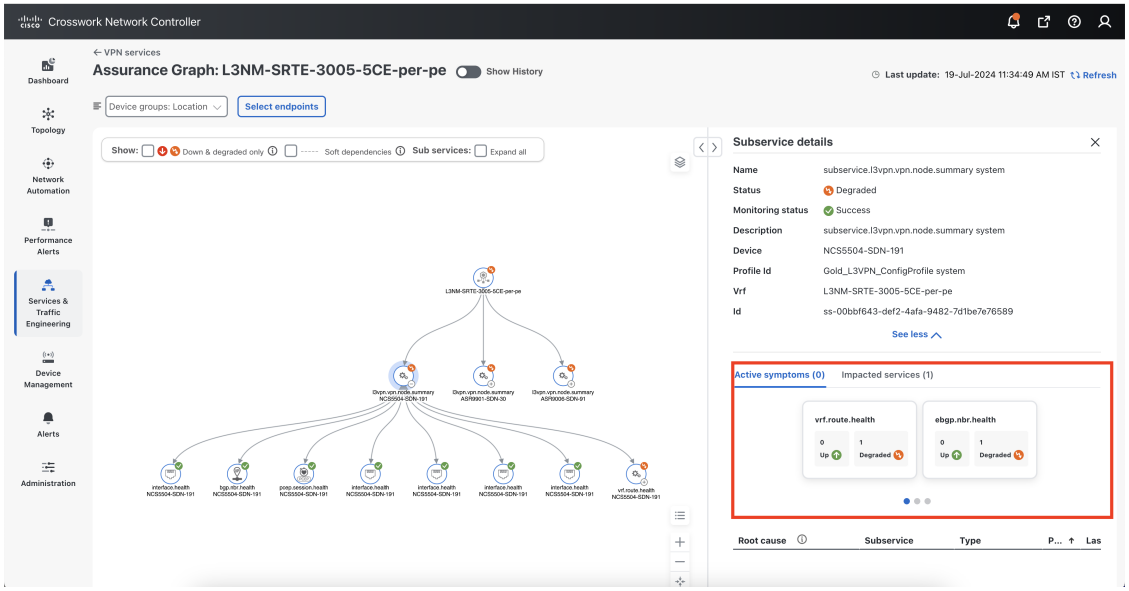
The Service details will continue to show the total symptoms count of the service, for the selected EOS time.

Note

If endpoints are selected, the total symptom count shows the combined number of symptoms for the selected endpoints.

Service details

In the **Service details** page, the **Active symptoms** tab shows the health details of feature-level nodes, including the number of subservices in the **Up** or **Degraded** state. Clicking on the **Degraded** state in a feature node filters the table to display symptoms and monitoring errors only for that node.



Use the Assurance Graph to identify root causes

Identify, inspect, and drill down to the root cause of a service degradation using the Assurance Graph and its subservice dependency view.

Before you begin

Ensure that Service Health monitoring is enabled for the service you want to inspect. Refer to Start Service Health monitoring for details.

Note

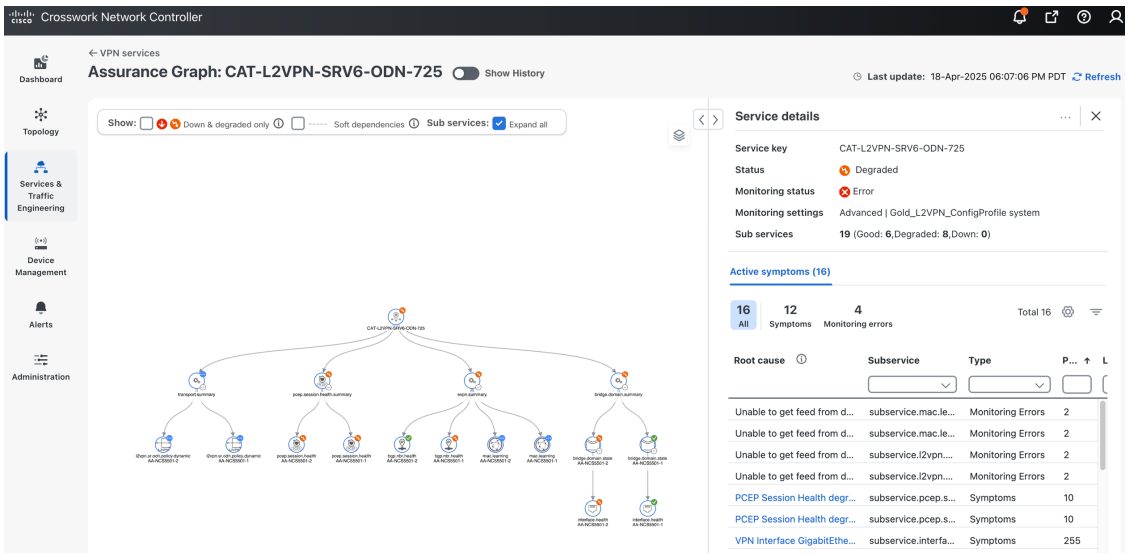
If you delete and then re-add a provider and devices for a monitored L3VPN service, the monitoring status remains degraded and displays as a monitoring error. To resolve this, stop and restart the service monitoring.

Identify, inspect, and drill down to the root cause of a service degradation.

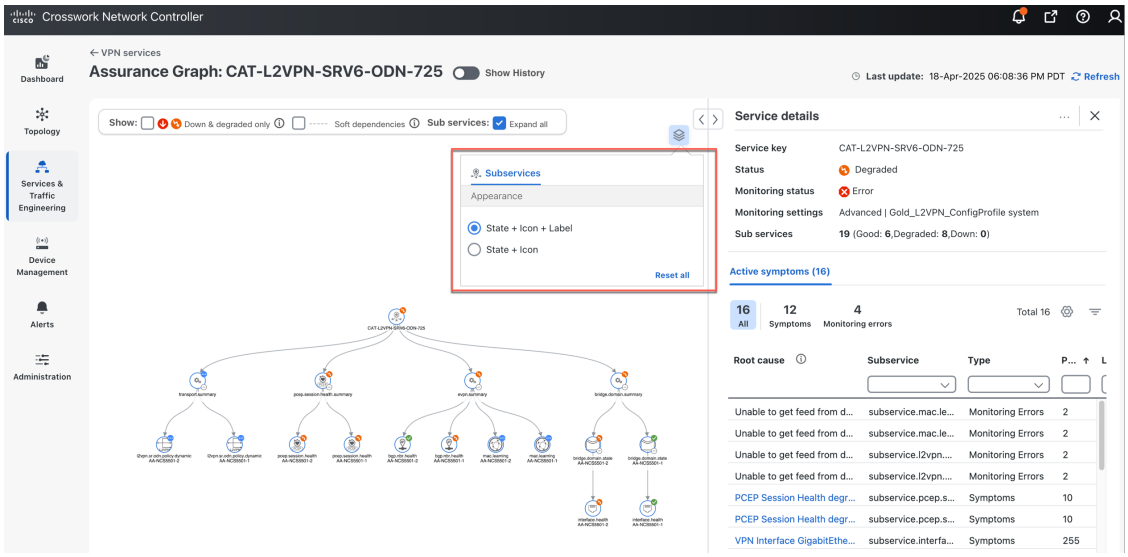
Procedure

1. From the main menu, choose **Services & Traffic Engineering** > **VPN Services**.
2. In the **Actions** column, click ⋯ for the required degraded service and click **Assurance graph**. The service assurance dependency graph displays services and subservices. The Service Details panel shows the Service Key, Status, Monitoring Status, Monitoring Settings, Subservices, and Active Symptoms details.

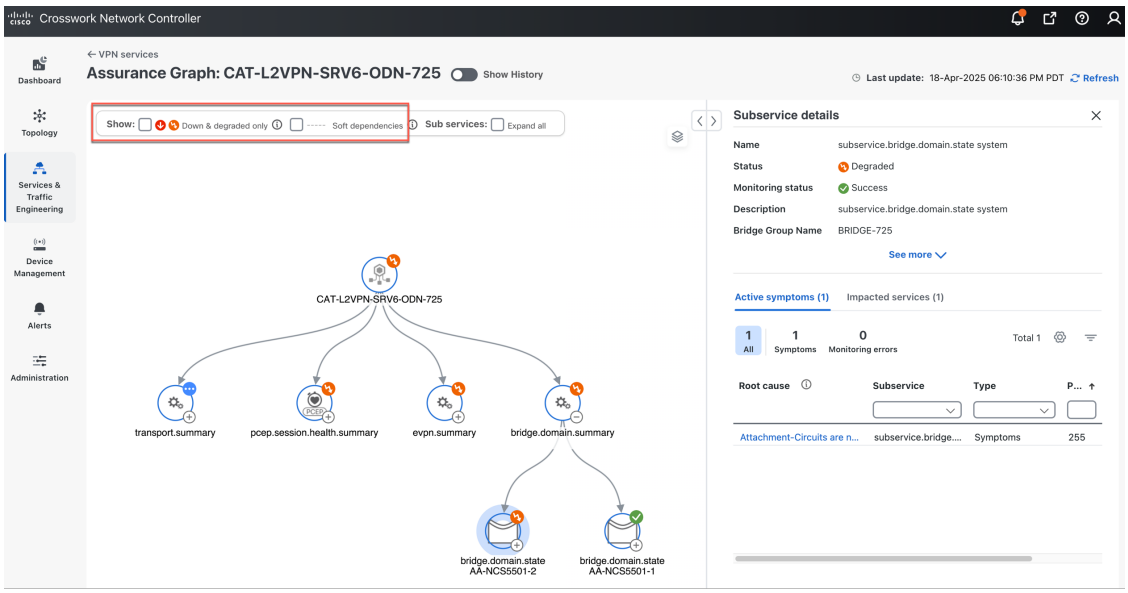
   This may take up to five to ten minutes to update after a service has been enabled for monitoring.



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**.



3. By default, the Assurance Graph displays a concise view with only the service and the top-level dependencies (feature nodes). Select the plus (**+**) icon in the nodes to expand the graph and to view the dependent details. To expand all the nodes at once, select **Sub services: Expand all** at the top.
4. Select a degraded subservice in the Assurance Graph. The Subservice Details panel shows subservice metrics, 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 mean that a child subservice health is weakly correlated with its parent health. Therefore, if the child health is degraded, the parent health is not affected.

5. Inspect the Active symptoms and Impacted services information, along with the root causes associated with the degraded service, to determine which issues require action to maintain a healthy setup.

What to do next

- To view the active symptoms and root causes, refer to Identify the active symptoms and root causes of a degraded service.
- To identify issues with degraded services within a specific time range, refer to Identify root causes using last 24 Hr metrics.
- To identify a service health issue by examining the collection jobs, refer to View the collection jobs.

Identify root causes using last 24 Hr metrics

Provides instructions for using last 24 Hr metrics and the Assurance Graph historical timeline to identify, analyze, and troubleshoot service degradation issues, inspect root causes and symptoms, and navigate relevant tools and visualizations in VPN services.

Before you begin

Ensure that service health monitoring is enabled for the service you want to analyze. Refer to Start Service Health monitoring for details.

> **Note**
>
> If you delete a provider and devices from a monitored L3VPN service and then add them again, the monitoring status remains degraded, displaying Monitoring error. To resolve this error, stop and restart service monitoring.

Use the last 24 Hr metrics to identify issues with degraded services during a specific time period. By isolating issues to a specific timeframe, you can examine details that may have caused a service to be degraded or down. This supports troubleshooting for the service or node and addresses symptoms.

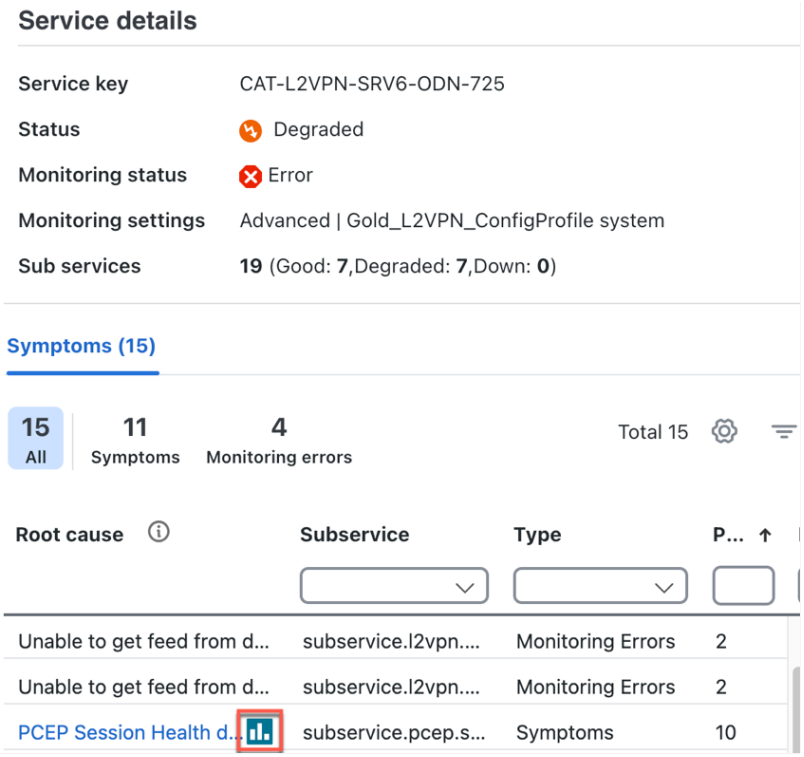
Procedure

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

   > **Note**
   >
   > This may take up to five to ten minutes to update after a service has been enabled for monitoring.

3. At the top of the page, click the **Show History** toggle. The historical timeline for the Assurance Graph appears showing ranges from the **Last Day** up to the **Last 60 Days**.
   a) To view data for a specific time range, select the range from the dropdown menu. Hovering over an event in the historical timeline displays a tooltip with event information.
   b) Review the root cause information by clicking a particular event. The Service details panel reloads, showing the active symptoms and the root causes associated with the event. You can resize columns using your mouse, or select the gear icon to choose which columns to display.

> **Note**
>
> Once you enable **Show History**, root cause information in the active symptoms table will start to show the blue last 24 Hr metrics icon. Data from the device will be initially delayed and may take some time before the last 24 Hr metrics begin to populate. Until then, zero is reported.



4. To further isolate the possible issues and to utilize the last 24 Hr metrics, perform these steps:
   a) In the historical timeline, use your mouse to select the range of historical health service monitoring details from one day (the **Last Day**) up to sixty days (**Last 60 Days**).

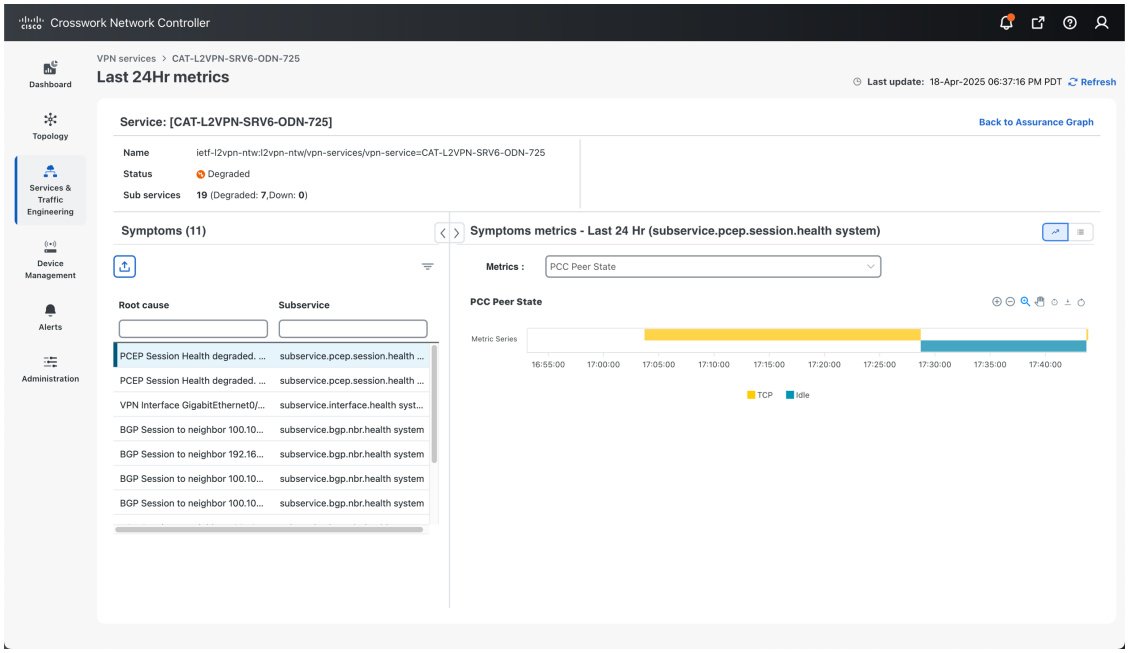
> **Note**
>
> At the top-right of the historical timeline, select the appropriate icons to either zoom in or out, and scroll horizontally through the date ranges. Or, refresh the graph to go back to the most recent event. You can also use your mouse to draw a rectangle over events to further zoom in on the degraded devices. Consecutive events may appear as a line of white space.

   b) Click a degraded event in the historical timeline. The Service details panel reloads, showing any active symptoms and the root causes to inspect. Expand the table as needed to view more information.

5. Check the **Down & degraded only** check box at the top-left corner of the Assurance Graph to display only degraded subservices. Inspect the Service details panel, which shows the active symptoms and their root causes. Uncheck the **Down & degraded only** checkbox and check the **Soft dependencies** check box in the top-left corner of the Assurance Graph. A soft dependency means that a child subservice's health has a weak correlation to its parent's health. As a result, degraded health in the child does not cause the parent's health to degrade.

Use the plus or minus symbols at the bottom-right corner of the Assurance Graph to zoom in or out on subservices. Select the **?** icon to view the link color legend.

6. Select the degraded subservice in the Assurance Graph to display its details.
7. Click the **Symptoms** tab to view any root causes for the Service Health details. Then, click the **Impacted services** tab to see the impacted services.
8. Click **X** in the top-right corner to return to the VPN Services list. In the Actions column, click ⋯ for the degraded service and click **Assurance graph** to show the Service details panel.
9. 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 showing metric patterns and different session states for individual root cause symptoms. It includes Status, Session, Start Time, and Duration information. Hover over the chart to view details.



What to do next

- To view the active symptoms and root causes, refer to Identify the active symptoms and root causes of a degraded service.
- To monitor VPN services using Assurance Graph capabilities and inspect degraded services or related nodes, refer to Use the Assurance Graph to identify root causes.
- To identify a service health issue by examining the collection jobs, refer to View the collection jobs.

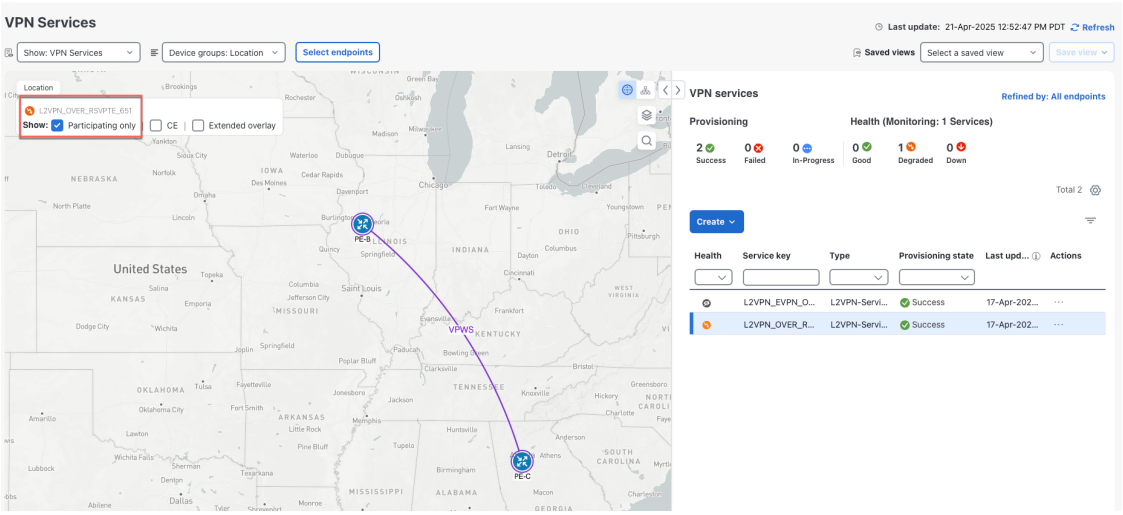
View the devices participating in the service

Guides users through viewing devices participating in VPN services using the topology view, identifying degraded devices and subservices, interpreting device badges, and determining appropriate steps to address unhealthy conditions within service assurance.

Use the topology view to see which devices participate in the services. You can also identify when a device or interface-related subservice degrades.

Procedure

1. From the main menu, choose **Services and Traffic Engineering** > **VPN Services**.
2. Click a service that shows as degraded. The topology map is updated, isolating the corresponding devices participating in that service.
3. At the upper left of the service assurance dependency graph view, select **Show: Participating only** so that the topology map shows only the devices participating in the service.



4. Hover your mouse over the device icons to view the popup information about Reachability State, Host Name, Node IP, and Type.

   A healthy device may display an orange badge when it has an underlying device or interface-related subservice that is not healthy. This highlights unhealthy subservices in the topological view, even when the device itself appears healthy. After examining Service Details for a device, you may find conditions such as low CPU on a subservice node. Identifying these conditions helps you determine steps to address the unhealthy subservice.

View the collection jobs

Outlines how to view and export parameterized collection jobs in Service Health, including steps for accessing job details, troubleshooting devices via Context ID, and exporting job status for further review using Crosswork Network Controller.

Use the **Parameterized jobs** tab on the Collection Jobs page to see all active jobs created by Service Health.

If Service Health is not deployed, this page will not contain any data.

Crosswork Network Controller enables you to view parameterized jobs. These jobs are template-based collection jobs that support many tasks, such as CLI collection jobs.

This feature is particularly useful for troubleshooting collection job issues. It allows you to examine the details of individual devices. Devices are identified by their Context ID, which is a protocol indicator. The Context ID specifies whether the jobs are gNMI, SNMP, or CLI-based.

Procedure

1. From the main menu, choose **Administration** > **Collection Jobs**.
2. Select the **Parameterized jobs** tab.

Review the parameterized jobs list to identify devices that may have service health degradation. Use the Context ID (protocol) to focus on gNMI, SNMP, and CLI-based jobs for additional troubleshooting.

3. In the Job details panel, select the collection job you want to export. Select  to download the status of collection jobs for further examination. The information is saved to a CSV file when you initiate the export.

   Note

   When exporting the collection status, you must enter the required information each time you execute an export. Make sure to review the **Steps to decrypt exported file** section on the Export Collection Status dialog box to access and view the exported data.

4. Select **Export**.
5. To check the status of the exported collection job data, select **View export status** at the top right of the Job details panel. The Export Status Jobs panel appears and provides the status of the export request.
6. Review the exported .csv file for collection job details and the possible cause of the degraded device.

5 Configure additional storage

Topics:

- Configure the additional external storage

This section explains these topics:

Configure the additional external storage

Configure AWS S3 external storage to store Service Health historical data.

Set up AWS S3 external storage in Crosswork Network Controller when prompted to ensure continued access to monitored data as internal storage approaches capacity.

Service Health provides internal storage of monitored data up to a maximum limit of 50 GB. The data includes the VPN service status at the time of storage and historical data about the service.

This data is stored by Service Health on your system in the tar.gz archive file format. Each tar.gz file represents an EOS. Service Health uses this data to display it visually in the Crosswork Network Controller UI when you click on an EOS.

When the storage reaches 70% capacity, Crosswork Network Controller generates an alarm prompting you to configure external storage. Service Health automatically deletes the least recently used files when 80% of the 50 GB storage capacity is reached.

When you configure external storage, most internal storage data is automatically moved to external cloud storage, while a portion remains locally in internal storage to serve as a cache for improved performance.

You can use an Amazon Web Services (AWS) cloud account to configure external storage in the cloud. Only AWS S3, an object storage service, is supported.

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

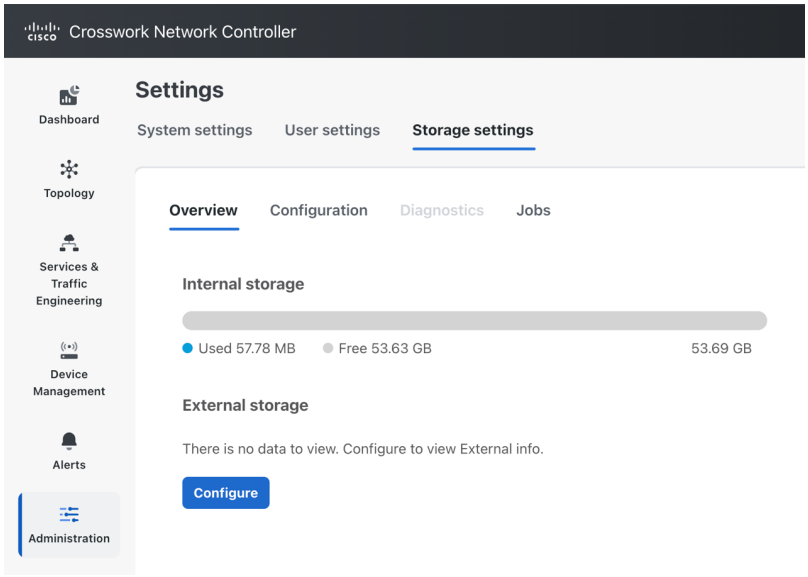
> **Note**
>
> External storage using Amazon Web Services (AWS) S3 is deprecated as of Release 7.2 and is no longer supported for storing Service Health historical data. This capability will be fully removed in Release 8.0.
>
> Service Health now uses internal storage, with a maximum capacity of 50 GB. We recommend to plan data retention and monitoring strategies accordingly.

Before you begin

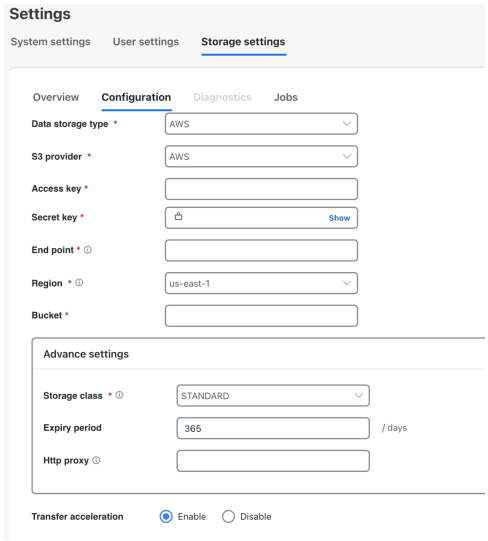
You must have an AWS cloud account set up to configure the external storage.

Procedure

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



2. With the Overview tab selected, select **Configure** under the **External storage** section. The Configuration screen appears, and the Data storage type and S3 provider fields are pre-populated with AWS.



3. Provide your AWS authentication information for all required fields, such as Access key, Secret key, Endpoint, and others.
4. Select **Copy local data** if you want all files previously stored in the local cache to be copied in bulk to external storage. This action enables incremental upload of 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 one (1), the files will be deleted two days later, at midnight in the operational time zone on the second day.

5. Click **Test and save**.
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 possible storage performance issues.

6 Customize Heuristic Packages

Topics:

- Heuristic packages
- Custom heuristic packages and rule mismatches
- Custom heuristic packages after Crosswork Network Controller upgrades

This section explains the following topics:

Heuristic packages

Introduces the core logic of Service Health using Heuristic Packages, outlining their components, rules, configuration profiles, subservices, and metrics, and guides users on accessing, viewing, and customizing heuristic packages for effective monitoring and reporting of various service types.

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 each service type.

To access the Heuristic Packages, select **Administration** > **Heuristic Packages** from the Main Menu.

The **Heuristic Packages** page has two tabs: **System** and **Custom**. The default set of Heuristic Packages provided with Service Health is called system packages. These packages are available in the **System** tab and cannot be modified. To modify a system package, export it, make your changes, and import it as a custom package that matches your preferences.

You can view the custom packages in the **Custom** tab.

Expand each section on this screen to see more details on monitored services and the thresholds used to generate alerts. For finer details and definitions, hover your mouse over the information **i** icon.

The Heuristic Package components and their functions are:

- **Rules**: Rules are used to structure services and the dependent subservices and metrics within a specific service type. Dependencies within these rules help define the subservices and the metrics that are required to generate the data used to assess the health of the service. A service can depend on an individual subservice, a list of subservices of the same type, or subservices of different types.
- **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.

  The system heuristics package in Service Health includes two configuration profiles, Silver and Gold for each service type (L2VPN and L3VPN). You can choose a profile option that aligns with your specific monitoring requirements. A Silver profile has more lenient thresholds than 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.

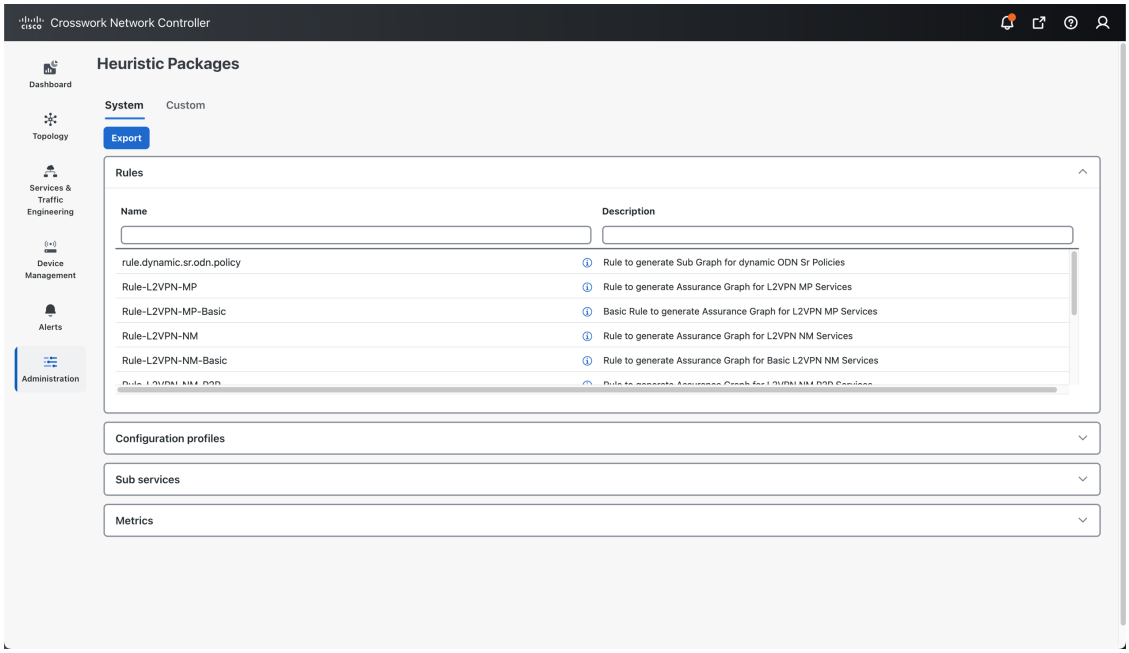
  For example, the subservice *subservice.evpn.health* monitors EVPN health. It is dependent on the metric *metric.l2vpn.xconnect.pw.state*. It evaluates an expression to check if *evpn_state* is **Up**. If the state is degraded, it raises a symptom.

  For the list of subservices supported in Service Health, refer to Supported VPN services with associated subservices on page 95.
- **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`.

Rules, Configuration Profiles, Sub services, and Metrics have a hierarchical relationship. Each Rule is mapped to a service type and depends on several subservices to compute service health. Subservices use metrics, and configuration profiles set threshold values for the metrics. Service Health assesses the health of the service using these values and then builds the Assurance Graph.

Figure 5: Service and subservice dependency



Customize the heuristic packages

The Heuristic Package 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. The Heuristic Package can adapt automatically to certain configuration variations, but other variations cannot be seamlessly absorbed. These variations include changes in the service function pack model, introducing a new device type in VPN service deployment, or adding new network features that require monitoring. In these scenarios, you may need to customize configuration profiles, rules, metrics, or subservice class definitions.

Refer to Create a custom heuristic package on page 63 for a basic example of how to create a custom package by customizing the configuration profile for a service. For details or help building a custom package based on rules, metrics, or subservices, contact the Cisco Customer Experience (CX) team or your Cisco account team.

Create a custom heuristic package

Creates a custom Heuristic Package by exporting the system package, editing the configuration profile JSON file, and compressing the updated folder for import.

Create a custom Heuristic Package and modify the `CPU_THRESHOLD_MAX` parameter in the `Gold_L2VPN_ConfigProfile` to set the maximum acceptable CPU usage for L2VPN services.

Procedure

1. From the main menu, choose **Administration** > **Heuristic Packages**. The Heuristic Packages page opens with **System** and **Custom** tabs.
2. Click the **System** tab and then click **Export**.

   The `exportAPI.tar.gz` package is downloaded to your system.
3. Extract the `exportAPI.tar.gz` file contents. This action provides the `system` folder.
4. In the `ConfigProfile` folder, open the `Gold_L2VPN_ConfigProfile-system.json` file.
5. Edit the file as described in these steps:
   a) Change the `namespace` attribute for the profile to `custom`.
   b) Increase the `version` attribute number to `2.0`.

c) Search for `CPU_THRESHOLD_MAX` and update the threshold value to eighty (`80`).

```
{
"id": "Gold_L2VPN_ConfigProfile system",
"name": "Gold_L2VPN_ConfigProfile",
  "namespace": "custom",
  "version": "2.0",
  "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 }
      }
    }
}
```

6. Save the file after making the changes.
7. Create a compressed tar.gz file from the **system** folder.

What to do next

Import the custom Heuristic Package into Crosswork Network Controller. Refer to Import the custom heuristic packages on page 65.

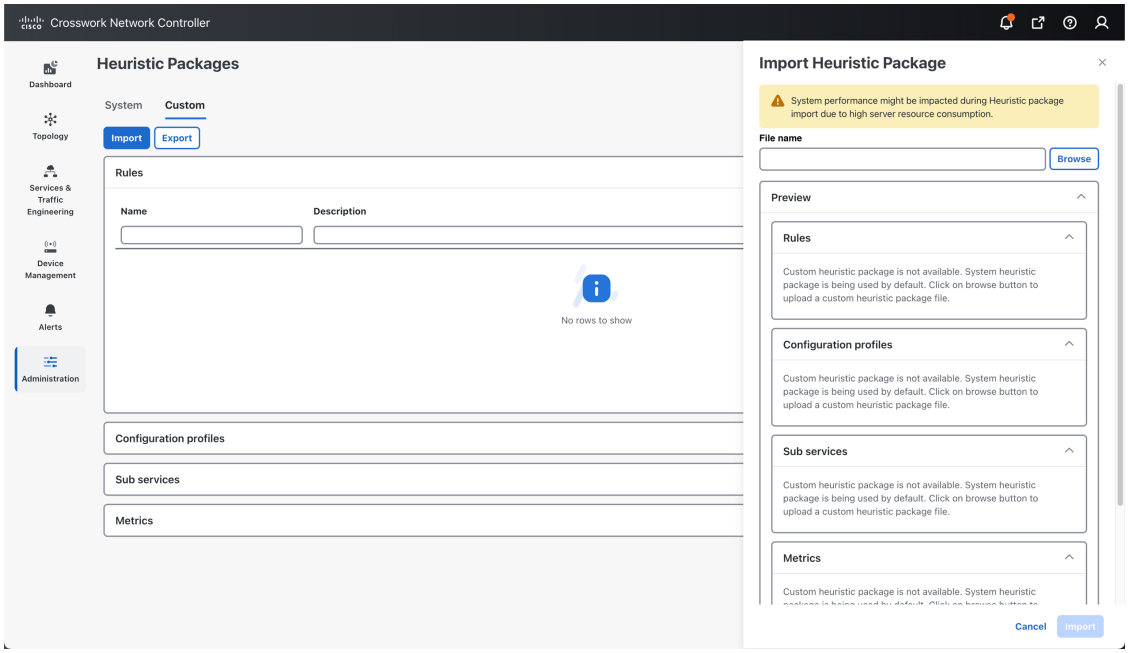
Import the custom heuristic packages

Imports a custom Heuristic Package into Crosswork Network Controller to replace the default system package for new monitoring sessions.

Import the custom Heuristic Package into Crosswork Network Controller.

Procedure

1. From the main menu, choose **Administration** > **Heuristic Packages**. The Heuristic Packages page opens with **System** and **Custom** tabs.
2. Click the **Custom** tab and then click **Import**. The **Import Heuristic Packages** panel appears.
3. Click **Browse** to locate the custom package archive file (*.tar.gz file) on your system.

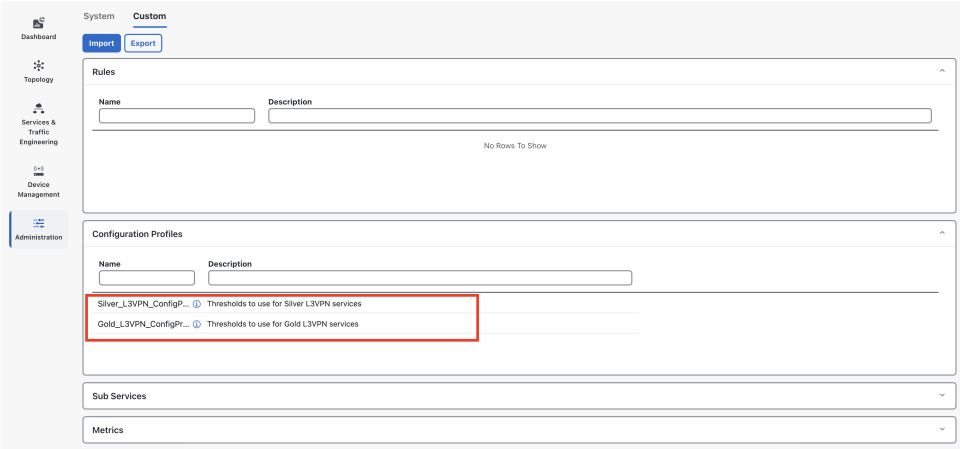


4. Select your custom package and click **Preview** to review the details of the package to be imported. Information for the package's Rules, Configuration Profiles, Sub Services, and Metrics appears.

> **Note**
>
> High server resource consumption during Heuristic Package import might impact your system performance.

Select each option to preview the details of the custom package. Crosswork Network Controller will validate the package and display an error message if it finds issues or a success message if validation is successful.

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



> **Note**
>
> When you import changes to a Heuristic Package by updating only the minor version, for example from 1.0 to 1.1, Service Health does not re-synchronize services that are already monitored or already processed with the earlier package. Those services continue using the previously applied package behavior. Minor-version changes take effect only for services that are monitored after the updated package is imported.
>
> To apply the updated Heuristic Package changes to existing monitored services, stop monitoring for those services and start monitoring again after the import.

What to do next

Stop monitoring the service before using custom heuristic packages. To resume monitoring, select the custom package and click **Start monitoring**. Refer to Start Service Health monitoring on page 20 for more information.

Custom heuristic packages and rule mismatches

Introduces custom heuristic package management, outlines the process for replacing default packages, details session handling to prevent rule mismatches, and provides guidance for managing session edits and monitoring errors in new and existing sessions.

When you upload a custom Heuristic Package, it immediately becomes the standard for all new monitoring sessions, replacing the default system Heuristic Package. To ensure monitoring consistency and prevent rule mismatches, you must manage your existing and new monitoring sessions after the upload.

Key actions and considerations:

- **Activation of custom Heuristic Package**:
  - All monitoring sessions started after the upload will automatically use the new custom Heuristic Package.
- **Handling existing monitoring sessions**:
  - Any monitoring session that began before the custom Heuristic Package upload will continue using the previous rules.
  - You must stop and restart these sessions to ensure they align with the new custom Heuristic Package.
- **Editing restriction**:
  - The Edit Monitoring Settings option does not function for sessions created with previous system Heuristic Packages.
  - Attempting to edit such sessions will result in a monitoring rule mismatch error and prompt you to restart the session.

After uploading, immediately identify and manually stop all monitoring sessions that began before the custom Heuristic Package upload.

Then, start new monitoring sessions to utilize the updated custom Heuristic Package. If you receive a monitoring rule mismatch error when editing, restart the session.

Example user workflow

1. Upload the custom Heuristic Package.
2. Stop any monitoring sessions that started before the upload.
3. Start new monitoring sessions (these will use the custom package).
4. If prompted by a mismatch error, restart the affected session.

> **Note**
>
> Always manage monitoring sessions promptly after uploading a custom Heuristic Package to maintain rule alignment and prevent errors.

Custom heuristic packages after Crosswork Network Controller upgrades

Outlines the need to manually update custom Heuristic Packages after Crosswork Network Controller upgrades, highlights impacted system changes to SR-PM and RSVP-TE subservices, and guides users on ensuring compatibility and access to updated metrics

Crosswork Network Controller supports both system and user-defined (custom) Heuristic Packages.

When you upgrade Crosswork Network Controller, your custom Heuristic Packages are not automatically updated with new system package changes. Manual updates are required to access the latest metrics and improvements.

In this release, these system Heuristic Package changes impact SR-PM (Segment Routing Performance Monitoring) and RSVP-TE (Resource Reservation Protocol-Traffic Engineering) subservices:

New metric classes

- heuristic_packages/MetricClass/metricSrPmRxPackets.json
- heuristic_packages/MetricClass/metricSrPmTxPackets.json
- heuristic_packages/MetricClass/metricMplsRsvpTeRxPackets.json

Updated subservices

- heuristic_packages/SubserviceClass/subserviceSrPolicyPccPmHealth.json
- heuristic_packages/SubserviceClass/subserviceSrPolicyPmHealth.json
- heuristic_packages/SubserviceClass/subserviceSrPolicyPcePmHealth.json

If you have imported or modified a custom Heuristic Package, review and manually update your package to include these new metric classes and subservice changes.

This ensures your customizations remain compatible and you continue to receive complete service health data for SR-PM and RSVP-TE features. For more information on creating custom Heuristic Packages, refer to Create a custom heuristic package on page 63.

A Reference - basic monitoring and advanced monitoring Rules

Topics:

- Basic and advanced monitoring rules, subservices, and metrics

This section explains the following topics:

Basic and advanced monitoring rules, subservices, and metrics

Service Health monitoring in Cisco Crosswork Network Controller provides two distinct monitoring levels—Basic and Advanced—that enable scalable and detailed service performance assessment. These monitoring rules leverage Heuristic Packages to evaluate service components and metrics, empowering network operators to maintain service integrity and quickly identify issues impacting network services.

Table 5: Basic and advanced monitoring rules for L2VPNs

Rule name (type)	Monitoring functionality	Metrics and subservices
Rule-L2VPN-NM- Basic	• Checks the health of the VPWS xconnect state. • Monitors the policies deployed by the ODN. • Monitors the health of the device: CPU and memory utilization.	subservice.l2vpn.sr.odn.policy.dynamic 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

	• Checks
	  • the health of the VPWS or EVPN xconnect state.
	  • BGP Neighbor session health.
	  • whether all BGP EVPN next hops for a given L2VPN service are reachable over LSP.
	  • path reachability between two endpoints.
	  • whether LSP path exists (in default VRF) towards the given destination device.
	• Health check for interface metrics: Oper status, interface in/out error packets, interface in/out packet discard.
	• Monitors
	  • the policies deployed by the ODN.
	  • the health of the device: CPU and memory utilization.
	  • the delta between received and transmitted packets between VPN interfaces and Pseudo-wire.
	  • Y.1731 probe stats for jitter, loss, and delay metrics, and compares against SLA thresholds.
	  • the health status of RSVP tunnel. Subservice health will be marked as 'degraded' in either of the below scenarios:
	  • FRR is configured, but backup is not ready.
	  • FRR backup is active (primary failed and traffic is flowing over FRR backup).
	• PCEP session state to all the peers configured on this device.
	• SR Policy (PCC initiated) health status. Admin should be up. Oper should be up. Oper should have
		subservice.l2vpn.sr.odn.policy.dynamic
		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
	stayed up since last polling.	metric.interface.in.errors metric.device.cpu.load metric.device.memory.free
Rule-L2VPN-NM- P2P-Basic	• Checks the health of the VPWS xconnect state. • Monitors the policies deployed by the ODN. • Monitors the health of the device: CPU and memory utilization.	subservice.l2vpn.sr.odn.policy.dynamic subservice.device.health subservice.vpws.ctrlplane.health

Rule-L2VPN-NM- P2P (Advanced)	• Checks
	  • the health of the VPWS xconnect state.
	  • the health for interface metrics: Oper status, interface in/out error packets, interface in/out packet discard.
	  • the SR Policy (PCC initiated) health status. Admin should be up. Oper should be up. Oper should have stayed up since last polling.
	• Monitors
	  • the policies deployed by the ODN.
	  • the health of the device: CPU and memory utilization.
	  • LSP path (in default VRF) towards the given destination IP address.
	  • the LSP path to the peer VPN node.
	  • path reachability between two endpoints.
	  • PCEP session state to all the peers configured on this device.
	  • Y.1731 probe stats for jitter, loss, and delay metrics, and compares against SLA thresholds.
		subservice.l2vpn.sr.odn.policy.dynamic
		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

Rule name (type)	Monitoring functionality	Metrics and subservices
Rule-L2VPN-MP- Basic	• 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

	• Checks BGP Neighbor health.
	• 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.
	• Groups
	  • together all the PCEP session health subservices.
	  • together all the device subservices.
	  • together all the bridge domain subservices.
	  • together all the transport subservices.
	• Monitors
	  • bridge domain state on a given endpoint.
	  • 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.
	  • the health of the device.
	  • the policies deployed by the ODN.
	  • PCEP session state to all the peers configured on this device.
	  • whether any routes are present for the given Bridge Domain.
	• SR Policy
	  • health status reflecting SR-PM SLA (if configured). Admin and Oper should be up. Oper should
		subservice.device.summary
		subservice.device.health
		subservice.pcep.session.health.summary
		subservice.pcep.session.health
		subservice.evpn.summary
		subservice.bgp.nbr.health
		subservice.mac.learning
		subservice.bridge.domain.summary
		subservice.bridge.domain.state
		subservice.interface.health
		subservice.transport.summary
		subservice.sr.policy.pcc.pm.health
		subservice.sr.policy.pce.pm.health
		subservice.mpls.rsvpte.tunnel.pm.health
		subservice.l2vpn.sr.odn.policy.dynamic
		metric.device.memory.free (supports XR only)
		metric.device.cpu.load (supports XR only)
		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.mac.learning.nexthops (supports XR only)
		metric.l2vpn.bridge.ac.state (supports XR only)
		metric.l2vpn.bridge.ac.list (supports XR only)
		metric.l2vpn.bridge.domain.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)

Rule name (type)	Monitoring functionality	Metrics and subservices
	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. ▪ 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. ▪ Subservice to reflect interface health.	metric.interface.in.discards (supports both XR and XE) metric.interface.out.discards (supports both XR and XE) metric.sr.policy.pcc.admin.state (supports XR only) metric.sr.policy.pcc.oper.state (supports XR only) metric.sr.policy.pcc.oper.up.time (supports XR only) metric.sr.policy.pm.delay.measurement (supports XR only) metric.sr.pm.delay (supports XR only) metric.sr.pm.variance (supports XR only) metric.sr.policy.pm.liveness.detection (supports XR only) metric.sr.pm.liveness.state (supports XR only) metric.sr.policy.pce.admin.state (supports XR only) metric.sr.policy.pce.oper.state (supports XR only) metric.sr.policy.pce.oper.up.time (supports XR only) metric.sr.policy.pce.ietf.policy.name (supports XR only) metric.sr.policy.pm.delay.measurement (supports XR only) metric.sr.pm.delay (supports XR only) metric.sr.pm.variance (supports XR only) metric.sr.policy.pm.liveness.detection (supports XR only) metric.sr.pm.liveness.state (supports XR only) metric.mpls.rsvpte.tunnel.oper.state (supports XR only) metric.mpls.rsvpte.tunnel.admin.state (supports XR only) metric.mpls.rsvpte.tunnel.frr.configured (supports XR only)

Rule name (type)	Monitoring functionality	Metrics and subservices
		metric.mpls.rsvpte.tunnel.frr.status (supports XR only) 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)

Table 6: Basic and advanced monitoring rules for L3VPN

Rule-L3VPN-NM- Basic	• Monitors the health of the device: CPU and memory utilization. • Reports the overall route connectivity health between the current PE device and its connecting CE device.	subservice.ce.pe.route.health subservice.device.health

Rule-L3VPN-NM (Advanced)	• Checks
	  • BGP Neighbor health.
	  • whether plain LSP route exists within given VRF towards given vpn ip-addresses.
	• eBGP Session health; Subservice to reflect interface health; Monitors the health of the device
	• 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.
	• 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.
	  • PCEP session state to all the peers configured on this device.
	• Subservice, together with child subservices in L3VPN Rule, reports the overall route health between current PE device and its connecting CE device.	

		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.nexthop.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
		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.rsvpte.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)
		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.1way.sd
		metric.l2vpn.y1731.pktloss.1way.ds
		metric.cef.route.labeled.lsp
		metric.cef.route.labeled.lsp
		metric.device.xconnect.ac.in.packets
		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.addrs
		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 metric.l2vpn.bridge.ac.list metric.l2vpn.bridge.domain.state
Rule-L3VPN-NM-VPN-Node-Basic	Monitors and presents the health summary for each PE device and the respective features underneath.	subservice.l3vpn.vpn.node.summary subservice.vrf.route.health subservice.device.health
Rule-L3VPN-NM-VPN-Node (Advanced)	Monitors and presents the health summary for each PE device and the respective features underneath.	subservice.l3vpn.vpn.node.summary subservice.vrf.route.health subservice.device.health subservice.interface.health subservice.pcep.session.health subservice.bgp.nbr.health

Service and subservice dependency example

The following example demonstrates how 'Rule-L2VPN-NM-P2P-Basic' interacts with its dependent subservices, 'subservice.vpws.ctrlplane.health' and 'subservice.device.health'. The detailed definitions of each subservice explain their metric dependencies and the resulting symptoms.

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_NSO",
      "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_NSO",
      "configSubSource": [
        "SUBSOURCE_SERVICE_CONFIG"
      ],
      "matchType": "MATCH_TYPE_XPATH",
      "matchExpression": "//flat-L2vpn/service-type[text()='p2p']",
      "matchPrefix": "",
      "matchParams": []
    },
    {
      "configSource": "SOURCE_TYPE_NSO",
      "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_NSO",
      "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_NSO",
```

```
      "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_NSO",
      "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"
}
```

```
{
  "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"
}
```

B Supported VPN services with associated subservices

The Service Health L2VPN and L3VPN types with their associated subservices provide a detailed, service-centric monitoring framework that benefits users by improving visibility, troubleshooting efficiency, and operational assurance for IOS XE and IOS XR devices in Crosswork Network Controller.

Supported VPN services with associated subservices for IOS XE devices

Supported VPN services	Associated subservices	Details
L2VPN P2P Plain	• Device Health • Path Reachability • Summary (aggregator) nodes • VPN Interface Health • Y.1731 Health	IOS XE does not support SNMP or gNMI collection type for this subservice (CEF route; XConnect). **Note** The reference to 'Plain' implies that L2VPN/L3VPN traffic takes the IGP path and does not use any transports, like SR.
L2VPN Point to Point over MPLS LDP	• Device Health • Path Reachability • Summary (aggregator) nodes • VPN Interface Health • VPWS Control Plane health • Y.1731 Health	IOS XE does not support SNMP or gNMI collection type for this subservice (CEF route; XConnect).
L2VPN Point to Point with SR underlay	• Device Health • Path Reachability • Summary (aggregator) nodes • VPN Interface Health • Y.1731 Health	IOS XE does not support SNMP or gNMI collection type for this subservice (CEF route; PCEP Session State; SRPolicy State; XConnect).

Supported VPN services	Associated subservices	Details
L3VPN SR	• BGP Neighbor Health (DynExp) • CE-PE Route Health • eBGP Neighbor Health • Path Reachability • Summary (aggregator) nodes • VPN Interface Health	IOS XE does not support SNMP or gNMI collection type for this subservice (CEF route; PCEP Session State). SR-ODN is also not supported.

Supported VPN services with associated subservices for IOS XR devices

Supported VPN services	Associated subservices

L2VPN EVPN Plain	• BGP
	  ◦ Neighbor Health (DynExp)
	  ◦ Nexthop Health (DynExp)
	• Device Health
	• EVPN Health
	• Path
	  ◦ Reachability
	  ◦ SLA
	• Plain LSP Path Health (DynExp)
	• Summary (aggregator) nodes
	• VPN Interface Health
	• VPWS Control Plane health
	Note
	The reference to 'Plain' implies that L2VPN and L3VPN traffic takes the IGP path and does not use any transports, like SR.

L2VPN EVPN SR	

Supported VPN services	Associated subservices

	• BGP
	  • Neighbor Health (DynExp)
	  • Nexthop Health (DynExp)
	• Device Health
	• EVPN Health
	• Fallback Enabled or Disabled (DynExp)
	• Path
	  • Reachability
	  • SLA
	• PCEP Session Health (DynExp)
	• SR Policy
	  • PCC
	  • PCE
	• Summary (aggregator) nodes
	• VPN Interface Health
	• VPWS Control Plane Health
	• Y.1731 Health

Supported VPN services	Associated subservices
L2VPN P2P Plain	• Device Health • Path Reachability • Path SLA • Plain LSP Path Health • Summary (aggregator) nodes • VPN Interface Health • VPWS Control Plane Health • Y.1731 Health **Note** The reference to 'Plain' implies that L2VPN and L3VPN traffic takes the IGP path and does not use any transports, like SR.
L2VPN Point to Point over MPLS LDP	• Device Health • Fallback Enabled/Disabled • Path Reachability • Path SLA • Summary (aggregator) nodes • VPN Interface Health • VPWS Control Plane Health • Y.1731 Health
L2VPN Point to Point over RSVP	• Device Health • Fallback Enabled/Disabled • Path Reachability • Path SLA • RSVP-TE Health • VPN Interface Health • VPWS Control Plane Health/Xconnect Health • Y.1731 Health

Supported VPN services	Associated subservices

L2VPN Point to Point with SR underlay	• Device Health
	• Fallback Enabled/Disabled
	• Path
	  • Reachability
	  • SLA
	• PCEP Session Health (DynExp)
	• SR Policy
	  • PCC
	  • PCE
	• Summary (aggregator) nodes
	• VPN Interface Health
	• VPWS Control Plane Health
	• Y.1731 Health

L3VPN SR	• BGP Neighbor Health (DynExp) • CE-PE Route Health • Device Health • eBGP Neighbor Health • Path Reachability • PCEP Session Health (DynExp) • SR and SRv6 polices • Summary (aggregator) nodes • VPN Interface Health • Vrf Plain LSP Path Health