



Cisco Network Insights for Resources Application for Cisco DCNM User Guide, Release 2.1.x

First Published: 2019-12-13

Last Modified: 2020-11-17

Americas Headquarters

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



CONTENTS

CHAPTER 1	New and Changed Information	1
	New and Changed Information	1

CHAPTER 2	Cisco Network Insights for Resources Installation	3
	About Cisco Network Insights for Resources	3
	Hardware Requirements	3
	Downloading Cisco Network Insights for Resources from the Cisco App Center	4
	Installing Cisco NIR Application in Cisco DCNM	5

CHAPTER 3	Cisco Network Insights for Resources Setup and Settings	7
	Cisco Network Insights for Resources Topology	7
	Cisco Network Insights for Resources Components in Cisco DCNM	8
	Guidelines and Limitations	9
	Cisco NIR App Initial Setup	10
	Cisco NIR App Settings	14
	Navigating Cisco NIR	15

CHAPTER 4	Using Cisco Network Insights for Resources	19
	Using the Cisco Network Insights for Resources Application	19
	Cisco NIR Dashboard	19
	Dashboard Inventory	19
	Dashboard Anomalies	20
	Cisco NIR System	22
	System Resource Utilization	22
	System Environmental	26
	Cisco NIR Operations	28

Statistics Analytics 28

Flow Analytics 30

CHAPTER 5**Supporting Third-Party Nodes for Cisco NIR Application 33**

About Third-Party Nodes Support for Cisco NIR Application 33

Third-Party Hardware Support for Cisco DCNM 33

Third-Party Nodes Limitations for Cisco NIR Application 33

Enabling Third-Party Nodes for Data Collection 34

Configuring Third-Party Nodes in Cisco DCNM 34

CHAPTER 6**Cisco NIR DCNM REST API Examples 35**

all_resources() 35

anomalies_details() 36

anomalies_summary() 37

flows_details() 37

flows_summary() 39

flows_top_flows() 40

flows_top_nodes() 42

get_fabrics_anomaly_summary() 43

get_fabrics_list() 43

get_nodes_list() 44

get_protocols_details() 45

get_protocols_resources() 46

get_protocols_topentities() 47

get_protocols_topnodes() 48

health_diagnostics() 49

service_health() 49

utilization_node_details() 50

utilization_top_nodes() 51

CHAPTER 7**Troubleshooting Cisco NIR App on Cisco DCNM 53**

Troubleshooting Cisco NIR Common GUI Issues 53

Troubleshooting at Cisco NIR App Level 54

Troubleshooting at Switch Level 54

Troubleshooting at Services Level	55
Troubleshooting at UTR Telemetry Receiver Level	56
Troubleshooting at Post-Processor Level	57
Troubleshooting at Event Collector, Predictor, and Correlator Services Level	58
Debugging Cisco NIR App on Cisco DCNM	59



CHAPTER

1

New and Changed Information

This chapter contains the following sections:

- [New and Changed Information, on page 1](#)

New and Changed Information

The following table provides an overview of the significant changes up to the current release. The table does not provide an exhaustive list of all changes or of the new features up to this release.

Table 1: New Features and Changed Behavior in the Cisco Network Insights for Resources application in Cisco DCNM for Release 2.1.x

Feature	Description	Release
UI enhancements	The UI enhancements in this release include cross launch navigation, the addition of top nodes in the dashboard, a fabric overview, and viewing of node details.	2.1.2
Cisco DCNM	Support for Cisco DCNM 11.3(1) release.	2.1.2
PC/vPC interface types	Support for PC/vPC interface types in the Interface Statistics tab.	2.1.2
BGP Statistics Telemetry	Support for BGP operational and statistical data in the Protocol Statistics tab.	2.1.2
Support for Flow Analytics	Flow Analytics allows to view flows and flow characteristics in each fabric.	2.1.2
Support for Flow Telemetry	Flow telemetry supports Cisco Nexus FX, FX2, and EX switches.	2.1.2
Third-party switch support	Support for third-party nodes for data collection.	2.1.2



CHAPTER 2

Cisco Network Insights for Resources Installation

This chapter contains the following sections:

- [About Cisco Network Insights for Resources, on page 3](#)
- [Downloading Cisco Network Insights for Resources from the Cisco App Center, on page 4](#)
- [Installing Cisco NIR Application in Cisco DCNM, on page 5](#)

About Cisco Network Insights for Resources

Cisco Network Insights for Resources (Cisco NIR) applications consist of monitoring utilities that can be added to the Cisco Data Center Network Manager (Cisco DCNM).

Hardware Requirements

This section describes the Cisco DCNM LAN deployment requirements for Cisco NIR software telemetry. A Cisco DCNM-native HA deployment is recommended.

The Cisco NIR application supports Cisco DCNM 11.3(1) release. It is recommended to use the latest Cisco DCNM release.

Table 2: Hardware Recommendations for Deployments up to 80 Switches and 2000 Flows

Node	Deployment Mode	CPU	Memory	Storage	Network
Cisco DCNM	OVA/ISO	16 vCPUs	32G	500G HDD	3x NIC
Computes (x3)	OVA/ISO	32 vCPUs	64G	500G HDD	3x NIC

Table 3: Hardware Recommendations for Deployments from 81 to 250 Switches and 10000 Flows

Node	Deployment Mode	CPU	Memory	Storage	Network
Cisco DCNM	OVA/ISO	16 vCPUs	32G	500G HDD	3x NIC

Node	Deployment Mode	CPU	Memory	Storage	Network
Computes (x3)	ISO	40 vCPUs	256G	2.4TB HDD	3x NIC*

* Network card: Quad-port 10/25G

The Cisco NIR application requires that physical servers hosting Cisco DCNM computes as VMs are at least Cisco C220-M4 category. It is also required that a compute be hosted on a data store with a dedicated hard disk of at least 500GB.

The following are hardware requirements for Cisco NIR app on Cisco DCNM.

1. Flow Telemetry is supported on the following series switches and line cards:

Cisco Nexus 9300-EX, -FX, and -FX2 platform switches.

2. Software Telemetry is not supported on the following series switches:

- Cisco N9K -GX and -FX3 platform switches.
- Cisco N3K-C3408-S, N3K-C3432D-S, N3K-C34200YC-SM, N3K-34180YC, and N3K-3464C switches.

Downloading Cisco Network Insights for Resources from the Cisco App Center

This section contains the steps required to download Cisco NIR app in the Cisco DCNM in preparation for installation.

Before you begin

You must have administrative credentials to download applications in Cisco DCNM.

-
- Step 1** Log in to the Cisco DCNM GUI with admin privileges.
 - If you don't have admin privileges, you can log in to the [Cisco App Center](#) to download the application.
 - Step 2** Choose **Applications**.
 - Step 3** Click **Browse App Center** on the far-right side of the screen.
 - Step 4** Search for Cisco Network Insights for Resources application on the search bar.
 - Step 5** Select the Cisco Network Insights for Resources application you want to download and click **Download** for that app to begin the process of downloading the app to your local machine.
 - Step 6** Review the license agreement and, if OK, click **Agree and download**.
The Cisco NIR app is downloaded to your local machine.
-

What to do next

Make sure the following requirements are met:

- Note the download location of the Cisco NIR app file on your local machine.
- Make sure the downloaded file can be accessed by the Cisco DCNM. If it cannot, move the file to a device and/or location where it can be installed on the Cisco DCNM.

Installing Cisco NIR Application in Cisco DCNM

This section contains the steps required to install Cisco NIR app in the Cisco DCNM.

Before you begin

Before you begin installing a Cisco NIR app, make sure the following requirements are met:

Step 1

You must have administrator credentials to install Cisco NIR application.

Step 2

You must have three compute servers installed and in the “Joined” state. For more information regarding the installation, discovery, and addition of compute servers, refer to the following sections:

- **Compute Installation:** For details on compute installation, refer to the [Installing a DCNM Compute](#) section.
- **DVS Security Settings:** For details on DVS security settings, refer to the [Networking Policies for OVA Installation](#) section.
- **Subnet Requirements for OOB and IB pool:** For details on subnet requirements for OOB and IB pool, refer to the [Subnet Requirements](#) section.
- **Creating a Compute Cluster:** For details on creating a compute cluster, refer to the [Enabling the Compute Cluster](#) section.
- **Adding Computers in Web UI:** For details on adding computers in web UI, refer to the [Adding Computes into the Cluster Mode](#) section.

What to do next

When the installation is complete, the application opens to a Welcome dialog where initial setup is performed. Continue with the setup of the Cisco NIR app located in the Initial Setup section of the next chapter.



CHAPTER 3

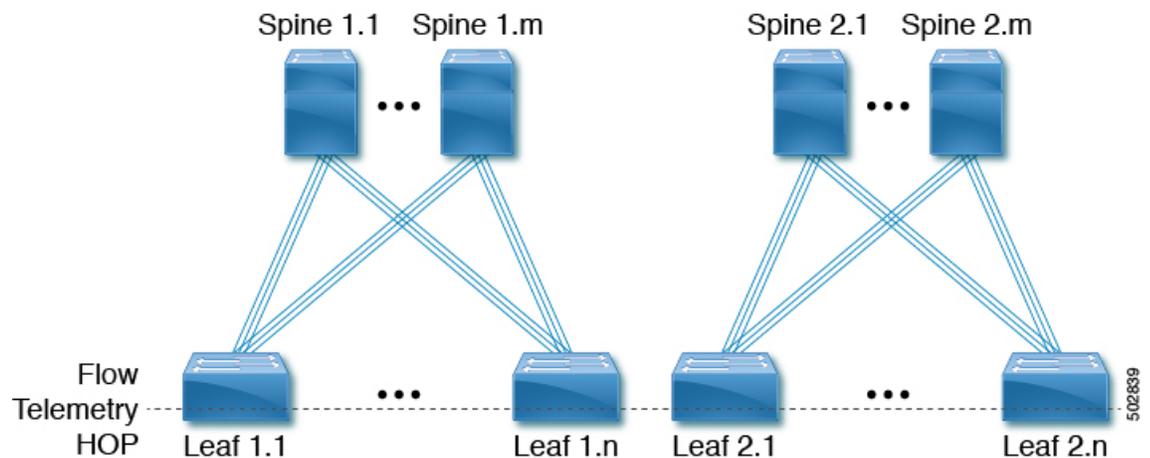
Cisco Network Insights for Resources Setup and Settings

This chapter contains the following sections:

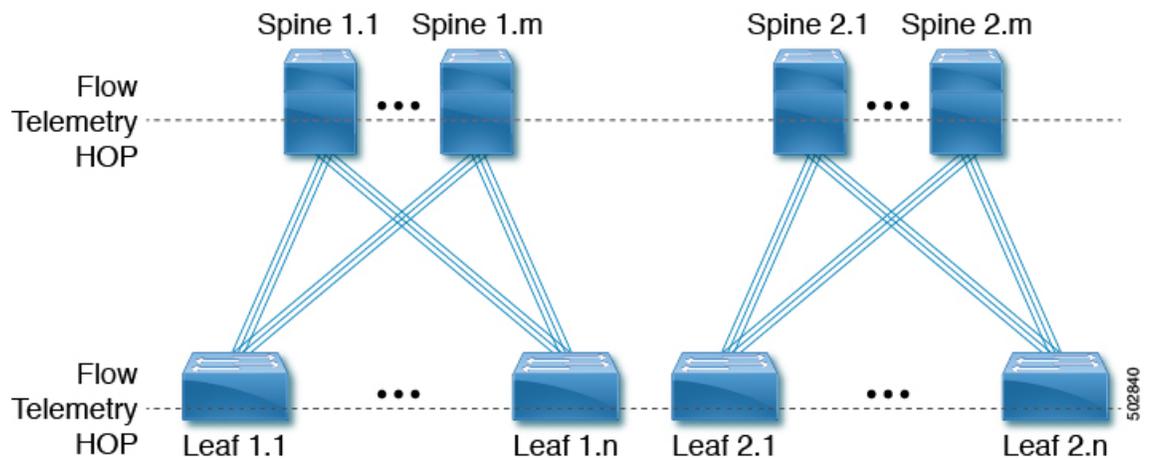
- [Cisco Network Insights for Resources Topology, on page 7](#)
- [Cisco Network Insights for Resources Components in Cisco DCNM, on page 8](#)
- [Guidelines and Limitations, on page 9](#)
- [Cisco NIR App Initial Setup, on page 10](#)
- [Cisco NIR App Settings, on page 14](#)
- [Navigating Cisco NIR, on page 15](#)

Cisco Network Insights for Resources Topology

The following figure describes the leaf switch-spine switch topology for 1-HOP or 2-HOP flow telemetry correlation.



The following figure describes the leaf switch-spine switch topology for 3-HOP or 4-HOP flow telemetry correlation.



Supported scenarios: The Cisco Network Insights for Resources topology supports the following scenarios.

VXLAN

- VPC on leaf switch
- Border spine switch
- Border leaf switch
- IR or Multicast underlay
- EBGp or IBGP
- IPv4 underlay
- IPv4 or IPv6 overlay

Legacy spine switch or leaf switch

- VPC on leaf switch
- IPv4 or IPv6

Cisco Network Insights for Resources Components in Cisco DCNM



The Cisco Network Insights for Resources (Cisco NIR) is a real-time monitoring and analytics application. The Cisco NIR app consists of the following components:

- **Data Collection**—The streaming of telemetry data is done by the Operating System on the fabric nodes. As each data source is different and the format in which data is streamed is different, there are corresponding collectors running analytics that translate the telemetry events from the nodes into data records to be stored in the data lake. The data stored in the data lake is in a format that the analytics pipeline can understand and work upon.

The following telemetry information is collected from various nodes in the fabric:

- **Resources Analytics**—This includes monitoring software and hardware resources of fabric nodes on Cisco DCNM.
- **Environmental**—This includes monitoring environmental statistics such as fan, CPU, memory, and power of the fabric nodes.
- **Statistics Analytics**—This includes monitoring of nodes, interfaces, and protocol statistics on Cisco DCNM and fabric nodes.
- **Flow Analytics**—This includes detecting anomalies in the flow such as average latency, packet drop indication, and flow move indication across the fabric.
- **Resource Utilization and Environmental Statistics**—Resource analytics supports configuration, operational and hardware resources. Environmental covers CPU, memory, temperature, and fan speed. System analytics also covers, Anomalies, and trending information of each resource and graphing of parameters which help Network operators to debug over period of time.
- **Predictive Analytics and Correlation**—The value-add of this platform is predicting failures in the fabric and correlating internal fabric failures to the user-visible/interested failures.
- **Anomaly Detection**—Involves understanding the behavior of each component while using different machine learning algorithms and raising anomalies when the resource behavior deviates from the expected pattern. Anomaly detector applications use different supervised and unsupervised learning algorithms to detect the anomalies in the resources and they log the anomalies in an anomaly database.

Guidelines and Limitations

The following are the guidelines and limitations for the Cisco Network Insights for Resources (Cisco NIR) application in the Cisco Data Center Network Manager (Cisco DCNM):

- After upgrading Cisco DCNM or Cisco NIR app to new version and before starting Cisco NIR app, make sure the following are set:
 - Navigate to **Applications > Preferences** from Cisco DCNM Configuration page and modify **Telemetry Network Configuration** to the desired value.



Note The `Out-Of-Band` is a default value for the interface, which may not be what you set prior to upgrade.

- Click **Submit**.

- When you did not modify the telemetry network configuration post Cisco DCNM or Cisco NIR app upgrade and enabled telemetry on any fabrics from Cisco NIR, then the application is not enabled and configured properly.
 - Login to Cisco DCNM active node using SSH client as `root`. In case you are already logged into Cisco DCNM active node, change to `root`.
 - Execute the following command.


```
curl -d '{"AppName": "NIR"}' http://127.0.0.1:9595/telemetry/force_cleanup_app
```
 - Execute the following command.


```
curl -d '{"AppName": "NIR"}' http://127.0.0.1:9595/telemetry/force_cleanup_hw_app
```
 - After few minutes all the fabrics in Cisco NIR Configuration page show a disabled state.
 - Once the status is disabled for all the fabrics, modify **Telemetry Network Configuration** in Cisco DCNM to the desired value.
 - Enable telemetry on the fabrics from the Cisco NIR Configuration page.
- For Flow Telemetry the Cisco NIR app captures the maximum anomaly score for a particular flow, for the entire cycle of the user specified time range.
- If Strict Config Compliance (SCC) is enabled in a fabric, you can not deploy Cisco NIR on Cisco DCNM.
- To enable telemetry on monitored fabric through Cisco NIR app, you must first delete all existing telemetry configurations on all the nodes in the monitored fabric before you enable this fabric from Cisco NIR app. The telemetry then assigns the receiver IPs to these nodes, which the Health page displays. The telemetry configuration will not push any telemetry configurations to the nodes because they are monitored. Therefore you have to check the receiver IPs from the Health page and must configure the nodes manually.
- After enabling telemetry in Cisco NIR app, to upgrade or downgrade a switch follow these steps:
 - Remove the switch from the fabric that needs upgrade or downgrade. Then upgrade or downgrade the switch image and add it back to the fabric.
 - Or, disable telemetry on the fabrics where switches need upgrade or downgrade. Then upgrade or downgrade the switch image and then enable telemetry on the fabrics.
- IPv6 is not supported for receiving telemetry data for Cisco NIR app.
- The Cisco NIR application requires that physical servers hosting Cisco DCNM computes as VMs are at least Cisco C220-M4 category. It is also required that a compute be hosted on a data store with a dedicated hard disk of at least 500GB. See [Hardware Requirements, on page 3](#).
- For instances where one or more fabrics do not recover from disabling state, you must stop and restart the Cisco NIR application in the Cisco DCNM. This will recover the failed disable state.

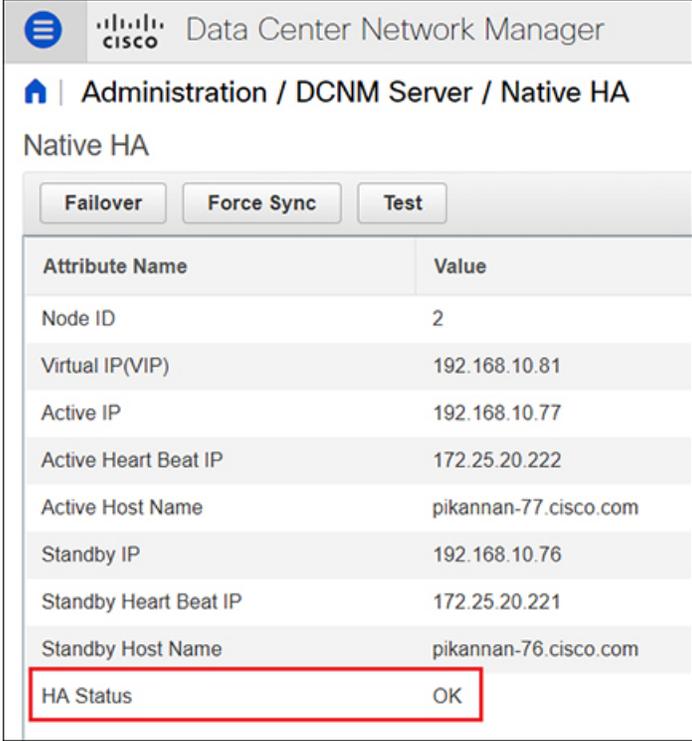
Cisco NIR App Initial Setup

The first time you launch the Cisco NIR app, you are greeted with a **Welcome to Network Insights** dialog. Follow these steps to complete the initial setup of Cisco NIR app:

Before you begin

Before you begin the initial set up of the Cisco NIR application in the Cisco DCNM, make sure the following prerequisites are met:

- The primary and standby hosts (HA) return a status of OK:
 1. In Cisco DCNM, click **Administration**.
 2. Under Cisco DCNM Server, click **Native HA**.
 3. Check the **HA Status** attribute as shown in the following image:



Attribute Name	Value
Node ID	2
Virtual IP(VIP)	192.168.10.81
Active IP	192.168.10.77
Active Heart Beat IP	172.25.20.222
Active Host Name	pikannan-77.cisco.com
Standby IP	192.168.10.76
Standby Heart Beat IP	172.25.20.221
Standby Host Name	pikannan-76.cisco.com
HA Status	OK



Note It may take some time for both hosts to be recognized. Once the OK status is displayed, AMQP notifications can begin. Check the AMQP server status below.

- The AMQP Server returns a status of OK:
 1. In Cisco DCNM, click **Dashboard**.
 2. In the **Server Status** tile, click **Health Check**.
 3. Check the status of the **AMQP Server** component as shown in the following image:

Server Status		
DCNM	Health Check	
Component	Status	Details
AMQP Server 10.126.216.240	✓ Ok	Service responds to user access.
DHCP Server Local	✓ Ok	Service is running.
TFTP Server Local	✓ Ok	Service is running.
EPLS Local	✗ Down	Service is stopped.
EPLC Local	✗ Down	Service is stopped.

- Precision Time Protocol (PTP) must be configured on all nodes you want to support with Cisco NIR. In both managed and monitor fabric mode, the user must ensure PTP is correctly configured on all nodes in the fabric. To ensure Precision Time Protocol is setup correctly:

- For details about Precision Time Protocol Easy Fabric, refer to [Precision Time Protocol for Easy Fabric](#).

Ensure PTP is enabled in Cisco DCNM easy fabric setup. The **Advanced** tab on Cisco DCNM fabric setup, check the box for **Enable Precision Time Protocol (PTP)**. For details, refer to [Add/Edit Fabric](#).

Step 1 On the welcome dialog, click **Begin First Time Setup**.

The **Network Insights Setup** window appears.

Step 2 On the **Network Insights Setup** window, click **Configure** to configure the **Data Collection Setup**.

The following steps enable the fabric to be monitored by Cisco NIR application.

Step 3 In the list of available fabrics, choose a fabric you want to monitor with Cisco NIR.

Step 4 In the **VXLAN / Classic** column, choose the fabric type:

- **VXLAN**: Identifies the fabric as a VXLAN fabric type.

Note If your network is a VXLAN fabric and you want to see VXLAN-specific information in the Cisco NIR application, you must select this option.

- **Classic**: Identifies the fabric as a Classic LAN fabric.

Step 5 In the **Mode** column, choose the mode you want to use for the fabric selected:

- **Managed**: Cisco DCNM monitors and manages the configuration of the nodes in the selected fabric. This option allows Cisco NIR app to push the telemetry configuration to the nodes in the chosen fabric.
- **Monitored**: Cisco DCNM does not deploy configuration to the nodes. Cisco DCNM discovers the nodes and displays them in the topology (read-only). Cisco NIR app will not send telemetry configuration to the nodes.

Note If this option is chosen, telemetry must be configured directly on the nodes in order for Cisco NIR app to receive data. The following configuration must be added on the NX-OS switches to stream telemetry data to Cisco NIR app when the fabric is configured to be in *Monitored* mode:

Example:

```

configure terminal

feature nxapi
feature ntp
feature lldp
feature icam
feature telemetry

telemetry
  destination-profile
    use-vrf management

  destination-group 500
    ip address <IP address of port 57500 protocol gRPC encoding GPB
  sensor-group 500
    data-source NX-API
    path "show vrf all" depth unbounded
    path "show nve vrf" depth unbounded
    path "show routing ip summary cached vrf all" depth unbounded
    path "show routing ipv6 summary cached vrf all" depth unbounded
    path "show ip mroute summary vrf all" depth unbounded
    path "show ipv6 mroute summary vrf all" depth unbounded
    path "show mac address-table count" depth unbounded
    path "show nve vni" depth unbounded
    path "show nve peers detail" depth unbounded
    path "show vlan summary" depth unbounded
    path "show vpc" depth unbounded
    path "show system internal icam app system internal access-list resource utilization" depth
unbounded query-condition show-output-format=json
    path "show system internal icam app hardware internal forwarding table utilization" depth unbounded
    query-condition show-output-format=json
  sensor-group 507
    data-source DME
    path sys/cdp depth 1 query-condition
    query-target=subtree&target-subtree-class=cdpIf, cdpAdjEp, cdpIfStats
    path sys/bgp depth 1 query-condition
    query-target=subtree&target-subtree-class=bgpEntity, bgpInst, bgpDom, bgpDomAf, bgpPeer, bgpPeerEntry, bgpPeerEntryStats, bgpPeerEvents, bgpPeerAfEntry

    path sys/lldp depth 1 query-condition
    query-target=subtree&target-subtree-class=lldpIf, lldpAdjEp, lldpIfStats
  sensor-group 508
    data-source DME
    path sys/intf depth 1 query-condition
    query-target=subtree&target-subtree-class=pcAggrIf&query-target-filter=deleted()
  sensor-group 503
    data-source DME
    path sys/intf depth 0 query-condition
    query-target=subtree&target-subtree-class=eqptFcotLane, eqptFcotSensor
  sensor-group 501
    data-source NX-API
    path "show port-channel summary" depth unbounded
    path "show lacp counters detail" depth unbounded
    path "show lacp interface" depth unbounded
    path "show lldp traffic interface all" depth unbounded
  sensor-group 502
    data-source DME
    path sys/intf depth 0 query-condition
    query-target=subtree&target-subtree-class=ethpmPhysIf, mmonEtherStats, mmonIfIn, mmonIfOut, ethpmAggrIf, llPhysIf, pcAggrIf

  sensor-group 505
    data-source NX-API
    path "show environment fan detail" depth unbounded

```

```

path "show environment power" depth unbounded
path "show system internal flash" depth unbounded
path "show clock" depth unbounded
path "show feature" depth unbounded
sensor-group 506
  data-source NX-API
  path "show system routing mode" depth unbounded
sensor-group 500
  data-source NX-API
  path "show module" depth unbounded
  path "show processes log" depth unbounded
  path "show icam scale" depth unbounded
  path "show environment temperature" depth unbounded
  path "show processes cpu" depth unbounded
  path "show processes memory physical" depth unbounded
  path "show system resources" depth unbounded
subscription 500
  dst-grp 500
  snsr-grp 504 sample-interval 61000
  snsr-grp 507 sample-interval 65000
  snsr-grp 508 sample-interval 0
  snsr-grp 503 sample-interval 62000
  snsr-grp 501 sample-interval 60000
  snsr-grp 502 sample-interval 60000
  snsr-grp 505 sample-interval 300000
  snsr-grp 506 sample-interval 3600000
  snsr-grp 500 sample-interval 59000

```

Step 6 Click **Save**.

Step 7 Click **Done**.

The second time you launch the Cisco NIR application, click **Review First Time Setup** to review the setup. Check **Do not show on launch** for the splash screen welcome dialog to not appear again.

Click **Get Started** to launch the application.

Cisco NIR App Settings

Once Cisco NIR app is installed, the following need to be checked off for the application to be fully set up:

- NTP and Time Zone Configuration

If there are Faults present in the application, they will show on the Faults tab. In the **Settings** menu click **Collection Status**, you should see the green circles in the table indicating the nodes where information is being transmitted.

Property	Description
Time Range	Specify a time range and the tables below display the data that is collected during the specified interval.
Fabric	Choose a fabric containing the nodes from which to collect telemetry data.

Property	Description
	<p>Clicking on this icon allows you to alter the following:</p> <ul style="list-style-type: none"> • Flow Collection Configuration—Enable or disable flow collection and choose a previously configured fabric. Create a VRF flow collection rule configuration per fabric: <ul style="list-style-type: none"> • Choose the Fabric from the drop-down. • Click the Plus icon and enter the VRF name. • Select the switch to create a flow collection rule. • Click Save. • System Status—Displays software, hardware, operational, and capacity usage of the Cisco NIR application on the compute cluster. • Collection Status—Displays data collection of System Metrics, and Events information per node. • NetworkInsights Setup—Lets the user configure the Cisco NIR application setup and enable or disable Flow Analytics. • About Network Insights—Displays the application version number.

The Flow Collection Configuration example.

Cisco NIR Service Instance Status

To view Cisco NIR app service instance status, exit the Cisco NIR app and click the gear image in the lower left corner of the Cisco NIR app icon in the Cisco DCNM application work pane.

Navigating Cisco NIR

The Cisco NIR app window is divided into two parts: the Navigation pane and the Work pane.

Navigation Pane

The Cisco NIR app navigation pane divides the collected data into three categories:

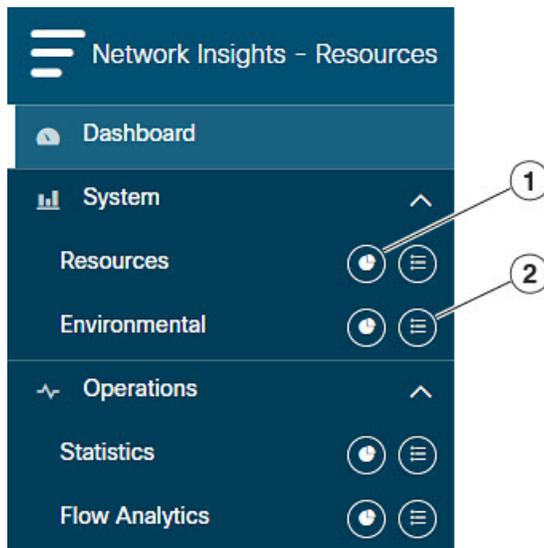


1 Dashboard: The main dashboard for the Cisco NIR app providing immediate access to anomalies.

2 System: Resource and environmental utilization.

3 Operations: Statistics information for interfaces and protocols.

Expanding System and/or Operations reveals additional functions:



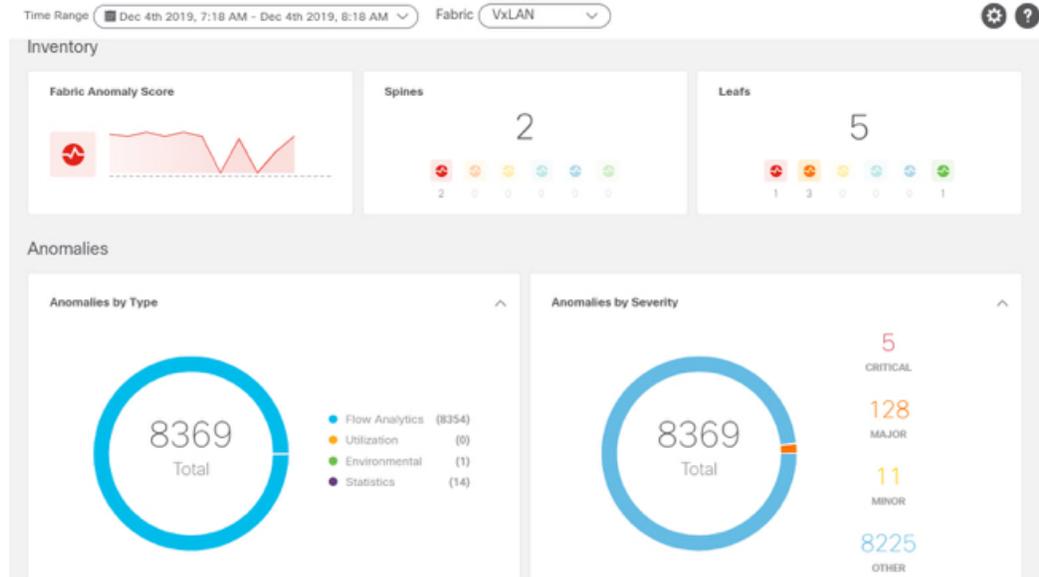
1 Dashboard View icon: Provides immediate access to top usage or issues for the selected telemetry type.

2 Browse View icon: Provides a detailed view of returned data for the selected telemetry type and allows for filtering to further isolate problem areas.

Work Pane

The work pane is the main viewing location in the Cisco NIR app. All information tiles, graphs, charts, and lists appear in the work pane.

Dashboard Work Pane



In an information tile, you can usually click on a numeric value to switch to the Browse work pane:



- 1 Launches the Browse work pane with all of the items displayed from the graph in the information tile.
- 2 Launches the Browse work pane with only the selected items displayed from the number in the information tile.

Browse Work Pane

The Browse work pane isolates the data for the parameter chosen on the Dashboard. The Browse work pane displays a top node lists, graphs over time, and lists all the nodes in an order defined by the anomaly score:

Start Time	End Time	Severity ^	Resource Type	Nodes	Description
May 16 2019 12:14:25pm	May 16 2019 07:54:37pm	⊗ Critical	config	N9Kv-2	Number of VRFs is above critical threshold (Usage : 991, Critical-Threshold : 900)
May 16 2019 12:14:53pm	May 16 2019 07:55:08pm	⊗ Critical	environmental	N9Kv-7	[Outlet Sensor] : Temperature is above critical threshold (Current Value : 75 C, Critical-Threshold : 72 C)
May 16 2019 12:14:17pm	May 16 2019 07:54:28pm	⊗ Critical	environmental	N9Kv-1	[Outlet Sensor] : Temperature is above critical threshold (Current Value : 75 C, Critical-Threshold : 72 C)

Clicking on one of the nodes in the list opens the Details work pane for that selection.

Details Work Pane

The Details work pane provides resource details about the item selected in the event list on the Browse work pane. The Details work pane consists of:

- General Information: Includes the anomaly score and the node name.
- Resource Trends: Includes operational resources, configuration resources, and hardware resources.
- Anomalies: Includes all anomalies for the node resource.



CHAPTER 4

Using Cisco Network Insights for Resources

This chapter contains the following sections:

- [Using the Cisco Network Insights for Resources Application, on page 19](#)

Using the Cisco Network Insights for Resources Application

Each node in the fabric streams telemetry data and events to a service in the Cisco NIR app. The Cisco NIR app analyzes the data and detects any anomalies. The Dashboards in the app provide relevant information to view.

Cisco NIR Dashboard

The Cisco Network Insights for Resources (Cisco NIR) application main dashboard provides immediate access to anomalies occurring in the network. Anomalies are learned deviations from the last known "good" state of a switch and are displayed by type and severity. Anomalies include resource utilization, environmental, and interface-level errors, and are color coded based on severity:

- Critical: Red
- Major: Orange
- Minor: Yellow
- Warning: Turquoise
- Information: Blue
- Healthy: Green

Dashboard Inventory

Anomalies are raised when a certain parameter threshold exceeds, or a rate of change threshold exceeds. The main dashboard displays the following information.

Property	Description
Fabric Anomaly Score	Displays the health of the fabric through color.
Spines	Displays the total number of spine nodes in the fabric with anomalies.

Property	Description
Leafs	Displays the total number of leaf nodes in the fabric with anomalies.

Click Spines and Leafs to view the details of the individual nodes in the fabric from Browse Nodes work pane.

Browse Nodes

The Browse Nodes pane displays the graph with top nodes based on Resource Utilization, Environmental, Statistics, End Point Analytics, and Flow Anamoly, which are various ways of viewing the behavior of the nodes. The page also dispalys the overview of the individual nodes in the fabric with node name, switch models, node type and other details. Click **Node** for the node detail view. The **Node Overview** section dispalys the top five nodes based on Resource Utilization, Environmental and Flow analytics with the break down of the faults and events. The **Anomalies** section displays the anomalies that the system detects.

The Browse Nodes pane displays the graph with top nodes based on Resource Utilization, Environmental, Statistics, and Flow Analytics, which are various ways of viewing the behavior of the nodes. The page also dispalys the overview of the individual nodes in the fabric with node name, switch models, node type and other details. Click the **Node** for the node summary pane to display all the gathered information for the selected node.

Click the  icon on the right top corner of the summary pane to show the **Node Details** page. The Node Details page displays General Information, Node Overview, and Anomalies. The **Node Overview** section dispalys the top five nodes based on Resource Utilization, Environmental, and Flow analytics with the break down of the faults and events. The **Anomalies** section displays the anomalies that the system detects.

On the detail page for the selected node, click the ellipses () icon on the right top navigation pane for additional related information for the node such as, Flows, Statistics, Resources, Anomalies, Events, and Environmental Resources.

From the ellipses menu click **Flows for the node** to open **Browse Flows** work pane, which filters the flows to view the top nodes by flow anomalies. Click **Statistics for the node** to open **Browse Statistics** pane, which filters the flows to view the top nodes by interface utilization.

Dashboard Anomalies

The main dashboard displays the anomalies detected in the fabric nodes.

Property	Description
Anomalies by Type	Displays the number of Anomalies by their type. Anomaly types include: <ul style="list-style-type: none"> • Flow Analytics • Utilization • Environmental • Statistics

Property	Description
Anomalies by Severity	<p>Displays the number of Anomalies (internal Fabric failures) and their severity level. Clicking on the area shows detail fault information, such as Node and Anomaly Score.</p> <ul style="list-style-type: none"> • Critical • Major • Minor • Info

Click any number from Anomalies by Type and Anomalies by Severity to access the Browse Anomalies work pane.

Browse Anomalies

The Browse Anomalies pane displays the graph with top nodes by anomaly score based on Type and Severity. The page also displays the overview of the individual nodes in the fabric with severity, resource type, node name, acknowledged and other details. Double-click the anomaly for the anomaly details. The **Anomaly Details** page displays the general information of the anomaly, anomalies, list of paths, and related details.

On the **Anomaly Details** page for the selected node, click the ellipses (⋮) icon on the right top navigation pane for additional related information for the node such as, Flows, Statistics, Resources, Anomalies, and Environmental Resources.

From the ellipses menu click **Flows for the node** to open **Browse Flows** work pane, which filters the flows to view the top nodes by flow anomalies. Click **Statistics for the node** to open **Browse Statistics** pane, which filters the flows to view the top nodes by interface utilization.

Browse Anomaly Filters

The Cisco Network Insights for Resources, application dashboard provides immediate access to anomalies occurring in the network. View, sort, and filter anomalies through the Browse Anomalies work pane.

You can refine the displayed anomalies by the following filters:

- Start Time - Display only anomalies with a specific start time.
- End Time - Display only anomalies with a specific end time.
- Description - Displays additional information about the anomaly.
- Node - Display only anomalies for specific nodes.
- Category - Display only anomalies from a specific category.
- Resource Type - Display only anomalies of a specific resource type.
- Severity - Display only anomalies of a specific severity.
- Acknowledged - Do not display the selected anomaly when checked to **T** for 20 minutes.

As a secondary filter refinement, use the following operators:

- = = - with the initial filter type, this operator, and a subsequent value, returns an exact match.

- != - with the initial filter type, this operator, and a subsequent value, returns all that do not have the same value.
- contains - with the initial filter type, this operator, and a subsequent value, returns all that contain the value.
- !contains - with the initial filter type, this operator, and a subsequent value, returns all that do not contain the value.

Property	Description
Start Time	The start time stamp for the anomaly detection.
End Time	The end time stamp for the anomaly detection.
Severity	The current severity level of the event. The levels are: <ul style="list-style-type: none"> • Critical—A service-affecting condition that requires immediate corrective action. For example, this severity could indicate that the managed object is out of service and its capability must be restored. • Major—Serious problems exist with one or more components. These issues should be researched and fixed immediately. • Minor—Problems exist with one or more components that might adversely affect system performance. These issues should be researched and fixed as soon as possible before they become a critical problem. • Other—Potential problems exist with one or more components that might adversely affect system performance if they are allowed to continue. These issues should be researched and fixed as soon as possible before they become a critical problem.
Resource Type	The resource type such as Flow , configuration, or operational.
Nodes	The node where the anomaly occurred.
Description	Additional information about the anomaly.

Cisco NIR System

The System section of the NIR application contains two areas of data collection:

- **Resource Utilization**—Fabric component capacity information.
- **Environmental**—Hardware component capacity information.

System Resource Utilization

The System Resources of the Cisco NIR application contains two areas of data collection.

Resource Utilization Dashboard

The Resource Utilization dashboard displays utilization, rate of change, trends, and resource anomalies over time for operational, configuration and hardware resources. Top leaf and spine nodes are displayed based on the factors that produced the high utilization.

Property	Description
Top Nodes by Capacity	The leaf node observations search can be more refined by filtering the information by the top leaf nodes.
Node Details	Displays the node trend observations by resource type: <ul style="list-style-type: none"> • Operational Resources • Configuration Resources • Hardware Resources

Browse Resource Utilization

View, sort, and filter statistics through the Browse Resource Utilization work pane.

Filters

You can refine the displayed statistics by the following filters:

- Node - Display only nodes.

As a secondary filter refinement, use the following operators:

- = - with the initial filter type, this operator, and a subsequent value, returns an exact match.
- != - with the initial filter type, this operator, and a subsequent value, returns all that do not have the same value.
- contains - with the initial filter type, this operator, and a subsequent value, returns all that contain the value.
- !contains - with the initial filter type, this operator, and a subsequent value, returns all that do not contain the value.

Property	Description
Top Nodes by	Displays the top nodes by: <ul style="list-style-type: none"> • MAC • IPv4 Host Routes • IPv6 Host Routes • IPv4 Prefix Routes • IPv6 Prefix Routes • Multicast Routes • VLAN • VRF • Port Usage • Ingress Port Bandwidth • Egress Port Bandwidth • CoPP • LPM • HRT • L2 QoS TCAM • L3 QoS TCAM • VTEP • VNI L2 • VNI L3 • VLAN • Ingress VLAN ACL • Egress VLAN ACL • Ingress Port ACL • Ingress Routed ACL • Egress Routed ACL

Property	Description
Operational Resources	Displays a list of operational resources based on anomaly score. List information includes: <ul style="list-style-type: none">• Anomaly Score• Node• MAC• IPv4 Host Routes• IPv6 Host Routes• IPv4 Prefix Routes• IPv6 Prefix Routes• Multicast Routes
Configuration Resources	Displays a list of configuration resources based on anomaly score. List information includes: <ul style="list-style-type: none">• Anomaly Score• Node• VLAN• VTEP• VNI<ul style="list-style-type: none">• L2• L3• VRF

Property	Description
Hardware Resources	<p>Displays a list of configuration resources based on anomaly score. List information includes:</p> <ul style="list-style-type: none"> • Anomaly Score • Node • Port Usage • Port Bandwidth • CoPP • LPM • HRT • QoS TCAM <ul style="list-style-type: none"> • L2 • L3 • VLAN ACL <ul style="list-style-type: none"> • Ingress • Egress • Port ACL <ul style="list-style-type: none"> • Ingress • Egress • Routed ACL <ul style="list-style-type: none"> • Ingress • Egress

System Environmental

The System Environmental of the Cisco NIR application contains two areas of data collection.

Environmental Dashboard

The Environmental Dashboard displays utilization, rate of change, trends, and anomalies over time for switch environmental resources such as fans, power, CPU, and memory.

Property	Description
Top Nodes by Utilization	Displays the percentage utilized per component: <ul style="list-style-type: none"> • CPU • Memory • Temperature • Fan Utilization • Power Supply • Storage
Node Details	Displays the node trend observations by environmental resource type.

Browse Environmental Resources

View, sort, and filter statistics through the Browse Environmental Resources work pane.

Filters

You can refine the displayed statistics by the following filters:

- Node - Display only nodes.

As a secondary filter refinement, use the following operators:

- == - with the initial filter type, this operator, and a subsequent value, returns an exact match.
- != - with the initial filter type, this operator, and a subsequent value, returns all that do not have the same value.
- contains - with the initial filter type, this operator, and a subsequent value, returns all that contain the value.
- !contains - with the initial filter type, this operator, and a subsequent value, returns all that do not contain the value.

Property	Description
Top Nodes by	Displays the top nodes by: <ul style="list-style-type: none"> • CPU (percent utilization) • Memory (percent utilization) • Temperature • Fan Utilization • Power Supply • Storage

Cisco NIR Operations

The Operations section of the Cisco NIR application contains two areas of statistical and analytical information:

- **Statistics Analytics**—Switch nodes interface usage and protocol statistics.
- **Flow Analytics**—Telemetry information collected from various devices in the Cisco ACI fabric to the NX-OS fabrics.

Statistics Analytics

The Operations Statistics section of the Cisco NIR application contains interface and protocol statistical information for top switch nodes.

Statistics Dashboard

The Statistics Dashboard displays top switch nodes by interface errors or usage, and protocol statistics.

Property	Description
Top Nodes by Interface Utilization	Displays the top nodes based on the combined bandwidth utilization of its interfaces.
Top Nodes by Interfaces	Displays the top nodes and lists the transmit and receive bandwidth utilization of each of its interfaces.

Browse Statistics Filters

Browse Statistics filters the interfaces to visualize the top interfaces by anomalies through the Browse Statistics work pane.

You can view, sort, and filter statistics through the Browse Statistics work pane. You can refine the displayed statistics by using the following filters:

- Node - Display only nodes.
- Interface - Display only interfaces.
- Protocol - Display only protocols.

As a secondary filter refinement, use the following operators:

- == - with the initial filter type, this operator, and a subsequent value, returns an exact match.
- != - with the initial filter type, this operator, and a subsequent value, returns all that do not have the same value.
- contains - with the initial filter type, this operator, and a subsequent value, returns all that contain the value.
- !contains - with the initial filter type, this operator, and a subsequent value, returns all that do not contain the value.

Property	Description
Top 10 Interfaces by	Displays the top interfaces by: <ul style="list-style-type: none"> • Transmit Utilization • Receive Utilization • Error
Interface Statistics	Displays a list of interface statistics based on anomaly score. List information includes: <ul style="list-style-type: none"> • Anomaly Score • Interface • Node • Type • Receive Utilization • Transmit Utilization • Errors
Protocol Statistics	Displays a list of protocol statistics based on anomaly score. List information includes: <ul style="list-style-type: none"> • Node • Type • Protocol • Number of Interfaces • Errors



Note In order for the Cisco NIR app to receive data from the nodes, confirm that all the nodes in the fabric are synced with PTP Grand master for hardware telemetry and NTP clock for software telemetry.

Browse Statistics

The Browse Statistics dashboard displays interface statistics and protocol statistics for the top interfaces by anomalies for nodes.

Interface Statistics

The Browse Statistics dashboard displays interface statistics for the top interfaces by anomalies for nodes that are of type - physical, port channel, and virtual port channel (PC and vPC) interfaces.

The green dot next to the interface name represents the operational status that the interface is active. The red dot next to the interface name represents that the interface is down.

The interface type is physical, port channel, or virtual port channel (PC or vPC) interface. Double-click **type** > **physical** for interface details of the node such as, node name, physical interface name, operational status, and admin state. The page also displays protocols, QoS, and DOM properties of the physical interface.

The port channel is an aggregate of physical interfaces and they can be statically channeled or can be dynamic using LACP protocols. The statistical data that collects the counters for packets, bytes and various errors are similar to that of physical interface. The sourceName differentiates the physical interface from port-channel (aggregated interfaces). The operational data is obtained by looking at additional set of objects that gives the admin-status, oper-status and list of member interfaces for both PC and vPC.

Click **type** > **pc** for interface details of the node such as, node name, port channel name, operational status, and admin state. The page also displays the anomalies, traffic, and member interfaces associated in the port channel.

The vPC is a logical interface that spans across two physical switches for fault tolerance. Double-click **type** > **vpc** for interface details of the node such as, node name, virtual port channel name, domain id, operational status, and admin state. The page also displays the anomalies, traffic, and the member interfaces associated in the nodes that are in the virtual port channel.

Protocol Statistics

The Browse Statistics dashboard displays protocol statistics for the top interfaces by anomalies for nodes that are of type CDP, LLDP, LACP, and BGP protocol. This page also displays node name and **Count** - the number of interfaces that the protocol is using or the number of sessions that the protocol is using for the node.

The BGP protocol data can be classified broadly into operational and statistical data. The operational data comprises of additional set of objects that gives the admin-status, oper-status and list of VRFs and VRF level information such as vrfName, vrfOperState, vrfRouteId, list of address family associated with each VRF, and list of peer and peer-entry information associated with each VRF. The statistical data comprises of peer-entry counters such as number of open's, updates, keepalives, route-refresh, capability, messages, notifications and bytes sent and received. It also includes peer-entry address family level the route count.

Double-click **protocol** > **BGP** for protocol details of the node such as, node name, protocol name, admin state, operational state and additional details. This page also displays the anomalies, neighbor nodes that are active, errors in the node, neighbor IP address, details about the established neighbors and not connected neighbors that the BGP protocol is using from the node family. Double-click a **Neighbor** node for the **Neighbor Details** window to popup with more details.

Double-click **protocol** > **CDP**, **protocol** > **LLDP**, or **protocol** > **LACP** for protocol details of the node such as, node name, protocol name, anomalies, interfaces that are active, errors in the node, and more details of the interface.

Flow Analytics

The Flow Analytics section of the Cisco NIR application displays the anomalies detected in the flow such as average latency, packet drop indication, and flow move indication collected from various nodes in the fabric.

Flow Analytics Overview

Flow Analytics provides deep insights at a flow level giving details such as average latency, packet drop indicator and flow move indicator. It also raises anomalies when the latency of the flows increase or when packets get dropped because of congestion or forwarding errors.

Each flow has a packet counter representing the number of packets entering the ASIC for that flow over a period of time. This period of time is called aggregation interval. There are several points where flow statistics for a given flow can be aggregated. Aggregation can happen in the ASIC, switch software, and server software.

Flow Analytics Hardware Requirements

Flow Analytics for Cisco NIR application in Cisco DCNM require the following hardware:

- For details on Flow Telemetry support for Cisco Nexus series switches, see [Hardware Requirements, on page 3](#).

Flow Analytics Limitations

The following are limitations for Flow Analytics for Cisco NIR application on Cisco Nexus EX line cards. For details on Flow Telemetry hardware support, see [Hardware Requirements, on page 3](#).

- Output port information for outgoing traffic from N9K-C93180YC-EX, N9K-C93108TC-EX, and N9K-C93180LC-EX line cards will not be displayed.

Flow Analytics Dashboard

The Flow Analytics Dashboard displays telemetry information collected from various devices in the fabric. The flow analytics records let the user visualize the flows in the fabric and their characteristics across the entire Cisco DCNM fabric.

Property	Description
Top Nodes by	The flow analytics engine also runs machine-learning algorithms on the behavior of the flows to raise anomalies in the behavior, such as Average Latency, Packet Drop Indicator, and Flow Move Indicator. The graph represents the anomalies in the behavior over a period of time.
Top Nodes by Flow Anomalies	Flow telemetry and analytics gives in-depth visibility of the data plane. The flow analytics engine collects the flow records streamed from the ASIC hardware and converts the 5-tuples to user-understandable flow records. Top nodes by flow anomalies displays the nodes in the network with the most anomalies. The details include, type of alarm, source destination, packet drops and latency.

In the **Top Nodes by Flow Anomalies** click the node card to display the Brows Flows page.

Browse Flows

The Browse Flows page displays the active nodes, ingress nodes, egress nodes, and flow collection filters, which display the anomalies in the fabric nodes.

Property	Description
Top 10 flows by	<p>Lists the top 10 flows that scored highest in the following:</p> <ul style="list-style-type: none"> • Anomaly Score—The score is based on the number of detected anomalies logged in the database. • Packet Drop Indicator—The flow records are analyzed for drops. The primary method of detecting drops is to check for discrepancies in the ingress and egress packet counts. • Latency—The time taken by a packet to traverse from source to destination in the fabric. <p>Note A prerequisite for fabric latency measurement is that all the nodes shall be synchronized with uniform time.</p> <ul style="list-style-type: none"> • Flow Move Indicator—The number of times a Flow moves from one Cisco DCNM leaf switch to another. The first ARP/RARP or regular packet sent by that endpoint appears as a flow entering the fabric through the new Cisco DCNM nodes.
Browse Flows	<p>Displays the following properties for the top nodes by flow anomalies:</p> <ul style="list-style-type: none"> • Anomaly Score—Displays the health state of the nodes. • Original Timestamp—Displays the timestamp when the anomaly occurred. • Nodes—Active nodes that show the anomaly score. • Ingress—Displays the Ingress switch name and VRF that show the flow anomalies. • Egress—Displays the Egress switch name and VRF that show the flow anomalies. • Source—Displays the VLAN, VNI, source address, and port information of the nodes by flow anomalies. • Destination—Displays the VLAN, VNI, destination address, and port information for the nodes by flow anomalies. • Address Type—Displays the address type as IPV4 or IPV6. • Protocol—Displays the protocol for the flow anomaly nodes. • Packet Drop Indicator—Displays the packet drops for the nodes. • Latency—Displays the latency information for the nodes. • Flow Move Indicator—Displays the packet flow moves for the nodes.

Double click the anomaly for the flow details. The **Flow Details** page displays the general information of the anomaly, anomalies, path summary, anomaly charts, and related details.



CHAPTER 5

Supporting Third-Party Nodes for Cisco NIR Application

This chapter contains the following sections:

- [About Third-Party Nodes Support for Cisco NIR Application, on page 33](#)
- [Third-Party Hardware Support for Cisco DCNM, on page 33](#)
- [Third-Party Nodes Limitations for Cisco NIR Application, on page 33](#)
- [Enabling Third-Party Nodes for Data Collection, on page 34](#)
- [Configuring Third-Party Nodes in Cisco DCNM, on page 34](#)

About Third-Party Nodes Support for Cisco NIR Application

The Cisco Network Insights for Resources app in Cisco DCNM provides a way to gather data from third-party nodes through Cisco NIR application. The data is acquired through the third-party collector service using REST based EAPI method calls provided by the collector service.

The following telemetry information is collected from third-party nodes in the fabric:

- **Environmental Statistics**—This includes monitoring environmental statistics such as CPU, memory, fan, temperature, and power usage, and storage details of the fabric nodes.
- **Interface Statistics**—This includes monitoring of nodes, interfaces, and protocol statistics on Cisco DCNM and fabric nodes using LLDP and LACP.
- **Resource Statistics**—This includes monitoring software and hardware resources of fabric nodes on Cisco DCNM using IPv4 unicast, IPv4 multicast, and MAC.

Third-Party Hardware Support for Cisco DCNM

The Cisco NIR app in Cisco DCNM supports Arista 7050SX and 7280SR Series switches.

Third-Party Nodes Limitations for Cisco NIR Application

The following are limitations for third-party nodes for Cisco NIR application.

- The Interface Statistics for LLDP and LACP do not support *Flap Count*, *Entries Aged Count*, and *PDU Timeout Count*.
- The Interface Statistics for MAC do not support local and static endpoints.
- Third-party nodes are supported only on Monitored mode.

Enabling Third-Party Nodes for Data Collection

Adding or removing the third-party nodes from the fabric will generate a control message, which triggers the third-party collector service present in the UTR pipeline to start or stop collecting data from the specific node.

To discover and enable third-party nodes to Cisco DCNM fabric:

- Create an external fabric to discover the third-party nodes, refer to [Creating an External Fabric](#) for details.
- To discover the third-party nodes, refer to [Discovering New Switches](#) for details.
- Add the third-party nodes to the external fabric, see [Adding non-Nexus Devices to External Fabrics](#) for details.

Configuring Third-Party Nodes in Cisco DCNM

Before you begin

Before you begin adding the third-party nodes to the fabric on Cisco DCNM, make sure the following requirement is met:

- You must have administrator credentials for doing the third-party node discovery.

Most of the Interface Statistics data is obtained with out any specific configuration for the third-party nodes. The following configuration is required for collecting port channel and storage statistics.

Step 1 Setup the port channel for LACP. See [Port Channel Configuration Procedures](#) for details.

Step 2 Execute the CLI command to collect storage statistics.

```
aaa authorization exec default local
```



CHAPTER 6

Cisco NIR DCNM REST API Examples

This chapter contains the following sections:

- [all_resources\(\)](#), on page 35
- [anomalies_details\(\)](#), on page 36
- [anomalies_summary\(\)](#), on page 37
- [flows_details\(\)](#), on page 37
- [flows_summary\(\)](#), on page 39
- [flows_top_flows\(\)](#), on page 40
- [flows_top_nodes\(\)](#), on page 42
- [get_fabrics_anomaly_summary\(\)](#), on page 43
- [get_fabrics_list\(\)](#), on page 43
- [get_nodes_list\(\)](#), on page 44
- [get_protocols_details\(\)](#), on page 45
- [get_protocols_resources\(\)](#), on page 46
- [get_protocols_topentities\(\)](#), on page 47
- [get_protocols_topnodes\(\)](#), on page 48
- [health_diagnostics\(\)](#), on page 49
- [service_health\(\)](#), on page 49
- [utilization_node_details\(\)](#), on page 50
- [utilization_top_nodes\(\)](#), on page 51

all_resources()

```
Get all resources .
REST URL      :
               GET /api/telemetry/utilization/resources.json
Parameters   :
               None
Example      :
               curl -k -i -XGET
               'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/utilization/resources.json'
Response     :
               {
                 "totalResultsCount": 5,
                 "totalItemsCount":5,
                 "entries": [
                   {
                     "categoryName": "",
```

```

        "resourceName": "EndPoints",
    }
    <-- SNIP LIST OF ALL OTHER RESOURCES -->
    {
    }
    ]
}

```

anomalies_details()

Get the anomalies in the system

REST URL :

```
GET /api/telemetry/anomalies/details.json
```

Parameters :

```

startTs (optional) => Start timestamp, default:now-1h
endTs (optional) => End timestamp, default:current-time
count (optional) => Num.of nodes in response, default:10
orderBy (optional) => Sort per the given field

```

Example :

```
curl -ksb -XGET
```

```
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/anomalies/details.json'
```

Response :

```

{
  "totalItemsCount": 90,
  "totalResultsCount": 90,
  "offset": 0,
  "entries": [
    {
      "anomalyId": "QUE0000000000018",
      "category": "System Resource",
      "startTs": "2018-09-19T16:45:05.679Z",
      "endTs": "2018-09-19T16:58:05.778Z",
      "entityName": "svc_ifc_policyelem",
      "severity": "critical",
      "anomalyType": "build-up",
      "nodeNames": [
        "leaf2"
      ],
      "resourceType": "queue",
      "resourceName": "recvQ",
      "anomalyStr": "[svc_ifc_policyelem] : Unexpected build-up of 7487 message[s]
in recvQ",
      "anomalyScore": 83
    },
    {
      "anomalyId": "QUE0000000000007",
      "category": "System Resource",
      "startTs": "2018-09-19T15:16:10.420Z",
      "endTs": "2018-09-19T16:49:01.289Z",
      "entityName": "svc_ifc_policyelem",
      "severity": "critical",
      "anomalyType": "build-up",
      "nodeNames": [
        "leaf1"
      ],
      "resourceType": "queue",
      "resourceName": "recvQ",
      "anomalyStr": "[svc_ifc_policyelem] : Unexpected build-up of 7502 message[s]
in recvQ",
      "anomalyScore": 83
    }
  ]
}

```

```
    ]
  }
```

anomalies_summary()

```
Get summary of the anomalies in the system
REST URL :
  GET /api/telemetry/anomalies/summary.json
Parameters :
  startTs (optional) => Start timestamp, default:now-1h
  endTs (optional) => End timestamp, default:current-time
Example :
  curl -ksb -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/anomalies/summary.json'
Response :
  {
    "totalAnomalyCount": 2,
    "totalAnomalyScore": 120.0,
    "entries": [
      {
        "severity": "warning",
        "anomalyCount": 1,
        "anomalyScore": 40.0
      },
      {
        "severity": "major",
        "anomalyCount": 1,
        "anomalyScore": 80.0
      }
    ]
  }
```

flows_details()

```
Get detailed flows
REST URL :
  GET /api/telemetry/flows/details.json
Parameters :
  startTs (mandatory) => Start timestamp,
  endTs (mandatory) => End timestamp, default:current-time
  filter (optional) => Lucene format filter
{srcIp,srcPort,dstIp,dstPort,ProtocolName,ingressVrf,egressVrf}, default:null
  statName (optional) => Stat name {flow:latency, flow:epmove, flow:pktdrop,
flow:ingressburstmax, flow:egressburstmax, flow:ingressPktCount, flow:egressPktCount}
  granularity (optional) => Granularity of time period
  fabricName (optional) => limit the records pertaining to this fabricName
Example:
  curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/flows/details.json'
Response:
  {
    "nodeName": null,
    "description": "",
    "statName": null,
    "entries": [
      {
        "flowId": "44.3.3.26:0",
        "srcIp": "44.3.3.26",
```

```

"dstIp": "42.2.2.22",
"srcPort": "0",
"dstPort": "0",
"protocol": "61",
"protocolName": "ANY-HOST",
"ingressVrf": "ctx4_1",
"egressVrf": "ctx4_1",
"flowType": "IPv4",
"ingressTenant": "tele4",
"egressTenant": "tele4",
"stats": [
  {
    "ingressPktCount": 6875,
    "ingressByteCount": 8250000,
    "egressPktCount": 0,
    "egressByteCount": 0,
    "ingressBurst": 0,
    "ingressBurstMax": 4800,
    "egressBurst": 0,
    "egressBurstMax": 0,
    "hashCollision": 0,
    "latency": 0,
    "srcMoveCount": 0,
    "dstMoveCount": 0,
    "moveCount": 0,
    "dropPktCount": 0,
    "dropNodes": [
      "telemetry-hw-spine1"
    ],
    "paths": [
      [
        {
          "node": "telemetry-hw-leaf3",
          "nodeType": "Leaf",
          "ingressVifs": [
            "eth1/1"
          ],
          "egressVifs": [
            "eth1/49"
          ]
        },
        {
          "node": "telemetry-hw-spine1",
          "nodeType": "Spine",
          "asicDropCode": 128,
          "dropReason": "",
          "dropType": "info",
          "ingressVifs": [
            "eth2/2"
          ],
          "egressVifs": [
            ""
          ]
        }
      ]
    ],
    "nodeName": [
      "telemetry-hw-leaf3",
      "telemetry-hw-spine1"
    ],
    "ingressNodes": [
      "telemetry-hw-leaf3"
    ],
    "egressNodes": [],

```

```

        "anomalyScore": 1,
        "dropReasons": [],
        "srcEpg": "testl3out",
        "dstEpg": "",
        "ts": "2019-02-01T19:18:56.458Z",
        "originTs": "2019-02-01T19:18:38.445Z",
        "terminalTs": "2019-02-01T19:20:42.419Z"
    }
],
"srcEpg": "testl3out",
"dstEpg": ""
}
]
}

```

flows_summary()

Browse flows.

REST URL :
GET /api/telemetry/flows/summary.json

Parameters :

- startTs (optional) => Start timestamp, default:now-1h
- endTs (optional) => End timestamp, default:current-time
- filter => Lucene format filter, default:null
- fabricName (optional) => limit the records pertaining to this fabricName

Example:

```
curl -k -i -XGET
```

```
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/flows/summary.json'
```

Response:

```

{
  "nodeName": null,
  "description": "",
  "statName": null,
  "entries": [
    {
      "flowId": "44.3.3.26:0",
      "srcIp": "44.3.3.26",
      "dstIp": "42.2.2.22",
      "srcPort": "0",
      "dstPort": "0",
      "protocol": "61",
      "protocolName": "ANY-HOST",
      "ingressVrf": "ctx4_1",
      "egressVrf": "ctx4_1",
      "flowType": "IPV4",
      "ingressTenant": "tele4",
      "egressTenant": "tele4",
      "stats": [
        {
          "ingressPktCount": 6875,
          "ingressByteCount": 8250000,
          "egressPktCount": 0,
          "egressByteCount": 0,
          "ingressBurst": 0,
          "ingressBurstMax": 4800,
          "egressBurst": 0,
          "egressBurstMax": 0,
          "hashCollision": 0,
          "latency": 0,
          "srcMoveCount": 0,
          "dstMoveCount": 0,
          "moveCount": 0,

```

flows_top_flows()

```

        "dropPktCount": 0,
        "dropNodes": [
            "telemetry-hw-spine1"
        ],
        "paths": [
            [
                {
                    "node": "telemetry-hw-leaf3",
                    "nodeType": "Leaf",
                    "ingressVifs": [
                        "eth1/1"
                    ],
                    "egressVifs": [
                        "eth1/49"
                    ]
                },
                {
                    "node": "telemetry-hw-spine1",
                    "nodeType": "Spine",
                    "asicDropCode": 128,
                    "dropReason": "",
                    "dropType": "info",
                    "ingressVifs": [
                        "eth2/2"
                    ],
                    "egressVifs": [
                        ""
                    ]
                }
            ]
        ],
        "nodeNameNames": [
            "telemetry-hw-leaf3",
            "telemetry-hw-spine1"
        ],
        "ingressNodes": [
            "telemetry-hw-leaf3"
        ],
        "egressNodes": [],
        "anomalyScore": 1,
        "dropReasons": [],
        "srcEpg": "test13out",
        "dstEpg": "",
        "ts": "2019-02-01T19:18:56.458Z",
        "originTs": "2019-02-01T19:18:38.445Z",
        "terminalTs": "2019-02-01T19:20:42.419Z"
    },
    "srcEpg": "test13out",
    "dstEpg": ""
}
]
}

```

flows_top_flows()

Get flows top flows.

REST URL :

GET /api/telemetry/flows/topFlows.json

Parameters :

startTs (optional) => Start timestamp, default:now-1h

endTs (optional) => End timestamp, default:current-time

granularity (optional) => Granularity of time period
 statName (optional) => Stat name {flow:latency, flow:epmove, flow:pktdrop}
 fabricName (optional) => limit the records pertaining to this fabricName

Example:

```
curl -k -i -XGET
```

```
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/flows/topFlows.json'
```

Response:

```
{
  "nodeName": null,
  "description": "",
  "statName": null,
  "entries": [
    {
      "flowId": "44.3.3.26:0",
      "srcIp": "44.3.3.26",
      "dstIp": "42.2.2.22",
      "srcPort": "0",
      "dstPort": "0",
      "protocol": "61",
      "protocolName": "ANY-HOST",
      "ingressVrf": "ctx4_1",
      "egressVrf": "ctx4_1",
      "flowType": "IPV4",
      "ingressTenant": "tele4",
      "egressTenant": "tele4",
      "stats": [
        {
          "ingressPktCount": 6875,
          "ingressByteCount": 8250000,
          "egressPktCount": 0,
          "egressByteCount": 0,
          "ingressBurst": 0,
          "ingressBurstMax": 4800,
          "egressBurst": 0,
          "egressBurstMax": 0,
          "hashCollision": 0,
          "latency": 0,
          "srcMoveCount": 0,
          "dstMoveCount": 0,
          "moveCount": 0,
          "dropPktCount": 0,
          "dropNodes": [
            "telemetry-hw-spine1"
          ],
          "paths": [
            [
              {
                "node": "telemetry-hw-leaf3",
                "nodeType": "Leaf",
                "ingressVifs": [
                  "eth1/1"
                ],
                "egressVifs": [
                  "eth1/49"
                ]
              },
              {
                "node": "telemetry-hw-spine1",
                "nodeType": "Spine",
                "asicDropCode": 128,
                "dropReason": "",
                "dropType": "info",
                "ingressVifs": [
                  "eth2/2"
                ]
              }
            ]
          ]
        }
      ]
    }
  ]
}
```

flows_top_nodes()

```

        ],
        "egressVifs": [
            ""
        ]
    }
]
],
"nodeNames": [
    "telemetry-hw-leaf3",
    "telemetry-hw-spine1"
],
"ingressNodes": [
    "telemetry-hw-leaf3"
],
"egressNodes": [],
"anomalyScore": 1,
"dropReasons": [],
"srcEpg": "test13out",
"dstEpg": "",
"ts": "2019-02-01T19:18:56.458Z",
"originTs": "2019-02-01T19:18:38.445Z",
"terminalTs": "2019-02-01T19:20:42.419Z"
}
],
"srcEpg": "test13out",
"dstEpg": ""
}
]
}

```

flows_top_nodes()

Get flows top nodes.

REST URL :

GET /api/telemetry/flows/topNodes.json

Parameters :

startTs (optional) => Start timestamp, default:now-1h

endTs (optional) => End timestamp, default:current-time

granularity (optional) => Granularity of time period

statName (optional) => Stat name {flow:latency, flow:epmove, flow:pktdrop},
default:flow-latency

fabricName (optional) => limit the records pertaining to this fabricName

Example:

curl -k -i -XGET

'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/flows/topNodes.json'

Response:

```

{
  "entries": [
    {
      "nodeName": "telemetry-hw-spine1",
      "description": "",
      "stats": [
        {
          "ts": "2019-02-01T19:16:32.002Z",
          "latency": 6
        }
      ]
    },
    {
      "nodeName": "telemetry-hv-leaf1",
      "description": "",
      "stats": [

```

```

        {
            "ts": "2019-02-01T19:16:32.002Z",
            "latency": 5
        }
    ]
}

```

get_fabrics_anomaly_summary()

```

Get fabric anomaly summary.
REST URL   :
            GET /api/telemetry/fabricsSummary.json
Parameters :
    fabricName (mandatory) => Name of the Fabric
    startTs           => Start timestamp, default:current-time - 1 hour
    endTs             => End timestamp, default:current-time
    include="anomalyScore" => Requires the Latest Maximum anomaly scores of the fabric,
    default:'no'
    history           => Requires the timeseries data of sum(anomaly scores, default:'no'
                        data, default=5m
                        granularity => applicable if history = "yes" , granulariry of the timeseries
Example     :
    curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/fabricsSummary.json'
Response    :
    {
        "anomalyScore" : "X"
        "entries": [
            {
                totalAnomalyScore ; X
                ts : now
            }
            .....
            {
                totalAnomalyScore ; X
                ts : now
            }
        ],
        "totalResultsCount": N,
        "totalItemsCount": N
    }

```

get_fabrics_list()

```

Get fabrics list.
REST URL   :
            GET /api/telemetry/fabrics.json
Parameters :
    filter           => Lucene format filter, default:null
Example     :
    curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/fabrics.json'
Response    :
    {
        "entries": [
            {

```

get_nodes_list()

```

        "fabricName": "FABRIC1",
        "fabricId": "1",
        "vendor": "CISCO_N9K_STANDALONE",
        "fabricType": "VXLAN",
        "configStatus": "ENABLED",
        "switchCount": 2,
        "controllerCount": 0
    },
    {
        "fabricName": "FABRIC2",
        "fabricId": "2",
        "vendor": "CISCO_ACI",
        "fabricType": "VXLAN",
        "configStatus": "ENABLED",
        "switchCount": 4,
        "controllerCount": 3
    },
    <--snip-->
],
"totalResultsCount": 11,
"totalItemsCount": 11
}

```

get_nodes_list()

```

Get nodes list.
REST URL   :
    GET /api/telemetry/nodes.json
Parameters :
    startTs (mandatory) => Start timestamp
    endTs       => End timestamp, default:current-time
    count       => Num.of nodes in response, default:1000
    filter      => Lucene format filter, default:null
Example    :
    curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/nodes.json'
Response   :
    {
        "entries": [
            {
                "nodeRole": "leaf",
                "nodeId": "302",
                "nodeName": "rleaf-scrimshaw2",
                "nodeMgmtIp": "1.2.3.4"
            },
            {
                "nodeRole": "spine",
                "nodeId": "205",
                "nodeName": "swmp14-dopplebock",
                "nodeMgmtIp": "1.2.3.4"
            },
            <--snip-->
        ],
        "totalResultsCount": 11,
        "offset": 0,
        "totalItemsCount": 11
    }

```

get_protocols_details()

```

Get Telemetry Protocol Stats details.
REST URL   :
            GET /api/telemetry/protocols/details.json
Parameters :
startTs    (mandatory) => Start timestamp
endTs      => End timestamp, default:current-time
fabricName => limit the records pertaining to this fabricName
nodeName   => Name of node
statName   => <protocol[:counter[:qualifier]], protocol[:counter[:qualifier]]...>

history    => '1' or '0', default is '0', indicates time-series request
granularity => Granularity of time period, default:5m
orderBy    => One statName of the format <protocol[:counter[:qualifier]]>
filter     => Lucene format filter to query for specific nodeName or sourceName,
default:null
Example    :
            curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/protocols/details.json'
Response   :
            {
              "totalResultsCount": 6,
              "totalItemsCount": 6,
              "offset": 0,
              "description": "Protocol statistical counters",
              "entries": [
                {
                  "nodeName": "leaf-103",
                  "entries": [
                    {
                      "sourceName": "phys-[eth1/14]",
                      "entries": [
                        {
                          "counterName": "InterfaceUtilisationIngress",
                          "value": 60.625,
                          "trending": "up",
                          "stats": [
                            {
                              "ts": "2018-10-24T05:05:00.000Z",
                              "value": 60.625
                            },
                            {
                              "ts": "2018-10-24T05:00:00.000Z",
                              "value": 59.827586206896555
                            },
                            {
                              "ts": "2018-10-24T04:55:00.000Z",
                              "value": 59.57142857142857
                            }
                          ]
                        }
                      ]
                    }
                  ]
                }
              ],
              <--snip-->
            {
              "sourceName": "phys-[eth1/11]",
              "entries": [
                {
                  "counterName": "LldpPktsEgress",
                  "value": 111.0,
                  "trending": "up",

```

get_protocols_resources()

```

        "stats": [
          {
            "ts": "2018-10-24T05:05:00.000Z",
            "value": 111.0
          },
          {
            "ts": "2018-10-24T05:00:00.000Z",
            "value": 110.10344827586206
          },
          {
            "ts": "2018-10-24T04:55:00.000Z",
            "value": 109.61904761904762
          }
        ]
      }
    ]
  }
}

```

get_protocols_resources()

```

Get Telemetry Protocol Stats resources.
REST URL   :
    GET /api/telemetry/protocols/resources.json
Parameters :
    filter           => Lucene format filter, default:null
    fabricName       => limit the records pertaining to this fabricName
Example    :
    curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/protocols/resources.json'
Response   :
    [
      {
        "protocol": "interface",
        "counter": "utilisation",
        "qualifiers": [
          "ingress",
          "egress"
        ]
      },
      {
        "protocol": "interface",
        "counter": "bytes",
        "qualifiers": [
          "ingress",
          "egress"
        ]
      }
    ]
<--snip-->
    [
      {
        "protocol": "lldp",
        "counter": "pkts",
        "qualifiers": [
          "ingress",
          "egress"
        ]
      },
      {
        "protocol": "lldp",

```

```

        "counter": "errors"
    }
]

```

get_protocols_topentities()

```

Get Telemetry Protocol Stats topEntities.
REST URL   :
            GET /api/telemetry/protocols/topEntities.json
Parameters :
    startTs (mandatory) => Start timestamp
    endTs    => End timestamp, default:current-time
    fabricName => limit the records pertaining to this fabricName
    statName  => parameter to find topEntities protocol[:counter[:qualifier]]
    granularity => Granularity of time period, default:5m
    filter    => Lucene format filter to query for specific nodeName or sourceName,
    default:null
Example    :
            curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/protocols/topEntities.json'
Response   :
    {
        "totalResultsCount": 6,
        "totalItemsCount": 6,
        "offset": 0,
        "description": "Protocol statistical counters",
        "entries": [
            {
                "nodeName": "leaf-103",
                "entries": [
                    {
                        "sourceName": "phys-[eth1/4]",
                        "entries": [
                            {
                                "counterName": "InterfaceUtilisationIngress",
                                "value": 65.53333333333333,
                                "trending": "down",
                                "stats": [
                                    {
                                        "ts": "2018-10-24T05:20:00.000Z",
                                        "value": 65.53333333333333
                                    },
                                    {
                                        "ts": "2018-10-24T05:15:00.000Z",
                                        "value": 65.78571428571429
                                    }
                                ]
                            }
                        ]
                    }
                ]
            },
            {
                "sourceName": "phys-[eth1/14]",
                "entries": [
                    {
                        "counterName": "InterfaceUtilisationIngress",
                        "value": 59.666666666666664,
                        "trending": "up",
                        "stats": [
                            {
                                "ts": "2018-10-24T05:20:00.000Z",
                                "value": 59.666666666666664
                            }
                        ],
                    }
                ]
            }
        ]
    }

```

get_protocols_topnodes()

```

        {
            "ts": "2018-10-24T05:15:00.000Z",
            "value": 59.5
        }
    ]
}

```

<---snip-->

get_protocols_topnodes()

Get Telemetry Protocol Stats topNodes.

REST URL :

GET /api/telemetry/protocols/topNodes.json

Parameters :

startTs (mandatory) => Start timestamp
endTs => End timestamp, default:current-time
fabricName => limit the records pertaining to this fabricName
nodeName => Name of node
statName => interface:utilization
summarize => '1' or '0', default is '0', summarizes across protocols

Example :

curl -k -i -XGET

'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/protocols/topNodes.json'

Response :

```

{
    "totalResultsCount": 6,
    "totalItemsCount": 6,
    "offset": 0,
    "description": "Protocol top nodes by score",
    "entries": [
        {
            "nodeName": "leaf-103",
            "entries": [
                {
                    "counterName": "protocol|utilization",
                    "stats": [
                        {
                            "ts": "2019-02-08T13:50:00.000Z",
                            "value": 62.33333333333336
                        },
                        {
                            "ts": "2019-02-08T13:45:00.000Z",
                            "value": 62.83333333333336
                        }
                    ]
                },
                {
                    "value": 62.33333333333336,
                    "trending": "down"
                }
            ]
        },
        ....
    ]
}

```

health_diagnostics()

```

Get health dianostics.
REST URL   :
            GET /api/telemetry/health/collectionStats.json
Parameters :
            None
Example    :
            curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/health/collectionStats.json'
Response   :
            {
              "totalItemsCount": 11,
              "entries": [
                {
                  "nodeName": "pod20-leaf3",
                  "stats": [
                    {
                      "resource": "sysStats",
                      "totalItemsCount": 9600,
                      "lastUpdatedTs": "2018-06-13T10:25:52.468Z",
                      "state": "HEALTHY"
                    }
                  ]
                }
              ],
              <---snip-->
            }

```

service_health()

```

Get the health of the services
REST URL   :
            GET /api/telemetry/health/serviceHealth.json
Parameters :
            None
Example    :
            curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/health/serviceHealth.json'
Response   :
            {
              "entries": [
                {
                  "serviceType": "THIRD_PARTY_SERVICE",
                  "serviceName": "elastic",
                  "state": "HEALTHY",
                  "displayName": "Data Store"
                },
                {
                  "serviceType": "CISCO_SERVICE",
                  "serviceName": "correlator",
                  "state": "HEALTHY",
                  "displayName": "Correlator"
                }
              ],
              <---snip-->
            }

```

utilization_node_details()

```

Get node details .
REST URL      :
    GET /api/telemetry/utilization/nodeDetails.json
Parameters   :
    None
Example      :
    curl -k -i -XGET
'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/utilizationnodeDetails.json'
Response     :
    {
        "totalResultsCount": 157,
        "totalItemsCount":157,
        "entries": [
            {
                "nodeName": "node-1",
                "entries": [
                    {
                        "resourceName":"cpu",
                        "latestValue":"85",
                        "maxValue":"100",
                        "resourceCategory":"",
                        "trending":"down",
                        "values":[
                            { "value":"85", "ts":"2018-02-21T20:21:03.109Z" },
                            {},
                            <--snip-->
                            {}
                        ]
                    },
                    {
                        "resourceName":"memory",
                        "latestValue":"84",
                        "maxValue":"100",
                        "resourceCategory":"",
                        "trending":"up",
                        "values":[
                            { "value":"84", "ts":"2018-02-21T20:21:03.109Z" },
                            {},
                            <--snip-->
                            {}
                        ]
                    },
                    <-- snip , LIST OF ALL OTHER RESOURCES -->
                    {
                        "resourceName":"ports",
                        "latestValue":"83",
                        "maxValue":"100",
                        "resourceCategory":"",
                        "trending":"up",
                        "values":[
                            { "value":"83", "ts":"2018-02-21T20:21:03.109Z" },
                            {},
                            <--snip-->
                            {}
                        ]
                    }
                ]
            },
            {
                "nodeName": "node-2"
            }
        ]
    }

```

```

    <-- same as in node-1 -->
  }
  <----snip LIST OF ALL OTHER NODES ---->
  {
    "nodeName": "node-10"
    <-- same as in node-1 -->
  }
]
}

```

utilization_top_nodes()

```

Get top nodes by utilization .
REST URL   :
            GET /api/telemetry/utilization/topNodes.json
Parameters :
            None
Example    :
            curl -k -i -XGET
            'https://<ip:port>/appcenter/Cisco/NIR/apiserver-api/api/telemetry/utilization/topNodes.json'
Response   :
            {
              "totalResultsCount": 10,
              "totalItemsCount":10,
              "entries": [
                {
                  "nodeName": "node-1",
                  "entries": [
                    {
                      "resourceName":"cpu",
                      "latestValue":"85",
                      "maxValue":"100",
                      "resourceCategory":"",
                      "trending":"down",
                      "values":[
                        { "value":"85", "ts":"2018-02-21T20:21:03.109Z" },
                        {},
                        <--snip-->
                        {}
                      ]
                    },
                    {
                      "resourceName":"memory",
                      "latestValue":"84",
                      "maxValue":"100",
                      "resourceCategory":"",
                      "trending":"up",
                      "values":[
                        { "value":"84", "ts":"2018-02-21T20:21:03.109Z" },
                        {},
                        <--snip-->
                        {}
                      ]
                    },
                    {
                      "resourceName":"ports",
                      "latestValue":"83",
                      "maxValue":"100",
                      "resourceCategory":"",
                      "trending":"up",
                      "values":[
                        { "value":"83", "ts":"2018-02-21T20:21:03.109Z" },

```

```
        {},
        <---snip-->
        {}
    ]
}
},
{
  "nodeName": "node-2"
  <-- same as in node-1 -->
}
<----snip---->
{
  "nodeName": "node-10"
  <-- same as in node-1 -->
}
]
}
```



CHAPTER 7

Troubleshooting Cisco NIR App on Cisco DCNM

This chapter contains the following sections:

- [Troubleshooting Cisco NIR Common GUI Issues, on page 53](#)
- [Troubleshooting at Cisco NIR App Level, on page 54](#)
- [Troubleshooting at Switch Level, on page 54](#)
- [Troubleshooting at Services Level, on page 55](#)
- [Troubleshooting at UTR Telemetry Receiver Level, on page 56](#)
- [Troubleshooting at Post-Processor Level, on page 57](#)
- [Troubleshooting at Event Collector, Predictor, and Correlator Services Level, on page 58](#)
- [Debugging Cisco NIR App on Cisco DCNM, on page 59](#)

Troubleshooting Cisco NIR Common GUI Issues

The following are troubleshooting tips for GUI issues on Cisco NIR app in Cisco DCNM .

- The Cisco NIR app has the ability to display historical data. The specific time duration can be selected from the available calendar to see data within that particular time range.
- The majority of issues will be due to receiving data from the APIs other than what was expected. Opening the **Developer Tools Network** tab and repeating the last action will show the API data received. If the issue is with the APIs, then troubleshooting will need to continue on the backend.
- If the API requests and responses are accurate then check the **Developer Tools Console** tab for any errors.
- After initial installation the application needs time to start. During this time, the GUI may exhibit incomplete or unstable behavior. It is recommended to wait several minutes before starting to use the application.
- Take screenshots just before and just after reproducing an issue. Screenshots along with a full network capture saved as HAR with contents can be used to issue reports. If an issue report has a HAR recording attached then there is a significantly higher chance that the root cause can be identified and resolved quickly.
- If the Cisco NIR GUI page loads to a skeleton template with spinner then this means almost none of the APIs are responding.
- If the Cisco NIR GUI page is taking a while to load fabrics then this means the `fabrics.json` API is not responding or not returning any fabrics.

- If the fabric anomaly score does not agree with reported anomalies, or the node counts are incorrect, then check the `fabricsSummary.json` response for the fabric `anomalyScore` value, and check the `nodes.json` response for the types and counts of nodes reported.
- If the expected fabrics are not shown in the fabric selection dropdown, first verify that they are not included in the `fabrics.json` response entries, then rerun setup and edit the data collection setup configuration to view the state of the configured fabrics. Make sure the appropriate fabrics are enabled and that no errors are reported. This data comes from the `get_nir_fabrics` request.
- For Flow Analytics issues make sure the following requirements are met:
 - The `capability.json` request is made when the GUI loads and returns true. If it returns false, it means the fabric does not support this feature.
 - Navigate to **Application Settings** tab and make sure **Flow Collection** has been enabled, the Management In-Band EPG has been selected, and verify the flow collection filters have been correctly configured.
 - To verify the MOs are using `visore`, navigate to **uni > fabric > flowcol** to check the configuration and check the classes `telemetrySelector`, `telemetrySubnetFltGrp`, and `telemetrySubnetFilter`.
 - Navigate to **Collection Status** tab and check if the nodes are returning flow telemetry.

Troubleshooting at Cisco NIR App Level

The following techniques are useful for troubleshooting Cisco NIR app on Cisco DCNM.

- To view running service instances, navigate to **DCNM > Application > NIR** and right-click the settings icon.
- To view capacity usage, navigate to **NIR**, right-click the settings icon and choose **System Status**.
- If there is no data shown in Cisco NIR dashboard, check the **Setup** page. Navigate to **NIR**, right-click the settings icon and choose **Rerun Setup**.
 - If no fabric is enabled, try to enable a fabric from available fabrics.
 - If no fabric is available, check if Cisco DCNM fabric is setup properly.
- To view telemetry data collection status, navigate to **NIR**, right-click the settings icon and choose **Collection Status**. Any red dot indicates that the telemetry data is not streamed. Possibilities for red dot are:
 - Switch is in monitor mode and telemetry configuration may not be available.
 - Switch telemetry configuration CLI may be missing or causing error.

Troubleshooting at Switch Level

The following techniques are useful for troubleshooting switches for Cisco NIR app on Cisco DCNM.

- To check the telemetry connection status:

- Login to the switch using SSH.
- Execute the following command and verify status of telemetry connection.

```
switch# show telemetry transport
```
- To check the telemetry data collector details:
 - Login to the switch using SSH.
 - Execute the following command

```
switch# show telemetry data collector details
```
- To check for any time sync issues from the switches:
 - Login to Cisco DCNM compute node using SSH.
 - Execute the following commands:

```
node# docker ps | grep debug
```

```
node# docker exec -it <<debugcontainerid>> sh
```

```
node# show ntp-time-sync
```

Check if the telemetry data is not synced with NTP server.

Troubleshooting at Services Level

The following techniques are useful for troubleshooting services for Cisco NIR app on Cisco DCNM.

- To check the basic services are running:
 1. To check for the required telemetry services are up and running:
 - Login to the master node and execute the following command.

```
node# docker service ls
```
 - Verify the services **apiserver**, **correlator**, **eventcollector**, **postprocessor**, **utr**, **redictor**, **scheduler**, **kafka**, **zookeeper**, and **elastic 6.1.4** are available and running.
 2. To check all the topics are pre-created and are in place:
 - Login to the compute node and execute the following command.

```
node# docker ps | grep debugtools
```
 - Get the container ID and execute the following command to debug the container.

```
node# docker exec -it <<debugcontainerid>> sh
```
 - Execute the following command in shell prompt.

```
sh# show kafka topics
```

- Verify the topics `cisco_nir-events`, `cisco_nir-operational`, `cisco_nir-stats-json`, and `cisco_nir-sw-telemetry-utr-out` are available.
3. To check all the indices are present:
 - Login to the compute node and execute the following command.


```
node# docker ps | grep debugtools
```
 - Get the container ID and execute the following command to debug the container.


```
node# docker exec -it <<debugcontainerid>> sh
```
 - Execute the following command in shell prompt.


```
sh# show elastic indices
```
 - Verify the indices `cisco_nir-enrich*`, `cisco_nir-statsdb*`, `cisco_nir-anomalydb`, `cisco_nir-fabrics*`, `cisco_nir-eventsdb`, `Cisco_nir-operdb`, `cisco_nir-recourcocollectdb*`, and `cisco_nir-recourcurescores*` are available.
 4. Other useful docker commands to check the basic services are running:
 - Login to the compute node and execute the following command.


```
node# docker ps
```
 - To get the memory and CPU statistics of containers running in compute node:


```
node# docker stats
```
 - Get the container ID to view the instant logs of any running container:


```
node# docker logs <<containerid>> -f
```
 - To view the logs of any running service in the master mode:


```
node# docker service logs <<servicecontainerid>> -f
```
 - To check the individual services are running:
 - Telemetry receiver stage.
 - Postprocessor stage.
 - Event collector, Predictor and correlator stage.

Troubleshooting at UTR Telemetry Receiver Level

The following techniques are useful for troubleshooting UTR telemetry receiver for Cisco NIR app on Cisco DCNM.

1. Make sure the Debug Tools app is available and running to debug a service.
 - Login to the compute node and execute the following command.

```
node# docker ps|grep debug
```

- Get the container ID and execute the following command to debug container.

```
node# docker exec -it <<debugcontainerid>> sh
```

2. Execute the following command to verify that the data is flowing through UTR service.

```
sh-4.2# show kafka data-utr-out --help
Usage: show kafka data-utr-out [OPTIONS]
```

Options:

```
-n, --node-name TEXT
-o, --offset [latest|earliest]
--help Show this message and exit.
```

3. Other useful commands to check UTR telemetry receiver data.

```
sh# show kafka data-utr-out -n leaf0 -o latest
```

```
sh# show kafka data-utr-out -n leaf0 -o latest | grep -i "show vlan
summary"
```

Troubleshooting at Post-Processor Level

The following techniques are useful for troubleshooting post-processor for Cisco NIR app on Cisco DCNM.

1. Make sure the Dubug Tools app is running to debug a service.

- Login to the compute node and execute the following command.

```
node# docker ps|grep debug
```

- Get the container ID and execute the following command to debug the container.

```
node# docker exec -it <<debugcontainerid>> sh
```

2. Execute the following command to verify that the data is flowing through post-processor service.

```
sh-4.2# show kafka data-processed --help
Usage: show kafka data-processed [OPTIONS]
```

Options:

```
-r, --stat-type [hardware|protocol|environmental|config|operational|interface|To be
removed]
-n, --node-name TEXT
-f, --fabric-name TEXT
-o, --offset [latest|earliest]
--help Show this message and exit.
```

3. Execute the following command to verify that the fabric node is flowing through post-processor service.

```
sh-4.2# show kafka data-fabricnodes --help
Usage: show kafka data-fabricnodes [OPTIONS]
```

Options:

```
-n, --node-name TEXT
-f, --fabric-name TEXT
-o, --offset [latest|earliest]
--help Show this message and exit.
```

4. Other useful commands to troubleshoot post-processor

```
sh# show kafka data-processed -r environmental -n leaf0 -o earlie
VXLAN-1
```

```
sh# show kafka data-processed -r environmental -n leaf0 -o earlie
VXLAN-1 | grep -l "powerDrawn"
```

Troubleshooting at Event Collector, Predictor, and Correlator Services Level

The following techniques are useful for troubleshooting collector, predictor, and correlator services for Cisco NIR app on Cisco DCNM.

- Execute the following command to verify the processed data is available in time series database.

```
sh-4.2# show nir es stats
Usage: show nir es stats [OPTIONS] COMMAND [ARGS]...
```

```
Options:
  --help Show this message and exit.
```

```
Commands:
  aggregated
  anomaly
  raw
```

```
sh-4.2# show nir es stats raw --help
Usage: show nir es stats raw [OPTIONS]
```

```
Options:
  -r, --stat-type [utilization|l2_protocol|interface]
  -n, --node-name TEXT
  -f, --fabric-name TEXT
  -s, --start-time [now-15m|now-1h|now-6h|now-1d|now-1w|now-1M]
  -p, --pretty
  --help Show this message and exit.
```

- Other useful commands to troubleshoot collector, predictor, and correlator services.

- Execute the following commands to check for the available fabrics.

```
# show nir es fabrics
# show nir es fabrics -f Simulation
```

- Execute the following commands to check for the available fabrics nodes.

```
# show nir es fabrics-nodes -f Simulation
# show nir es fabrics-nodes -f Simulation -n N9Kv-2
```

- Execute the following command to verify the Statistics data.

```
# show nir es stats raw -f Simulation -n N9Kv-1 -r utilization
```

- Execute the following command to verify the Aggregated data.

```
# show nir es stats aggregated -f Simulation -n N9Kv-2 -r config
```

- Execute the following command to verify the Anomaly data.

```
# show nir es stats anomaly -f Simulation -n N9Kv-1 -r config
```

Debugging Cisco NIR App on Cisco DCNM

The following techniques are useful for troubleshooting Cisco NIR app on Cisco DCNM.

- To fetch the software telemetry logs from the devices streaming the data execute the following command.

```
N9k-ToR2# show tech-support telemetry > ts_telemetry.log
```

- To fetch the logs for single node Cisco DCNM server execute the following command.

```
# appmgr afw fetch-logs Cisco:NIR
```

- To fetch the logs for Cisco DCNM server with HA (active-standby):

- Login to the active Cisco DCNM server.

- Execute the following command.

```
# appmgr afw fetch-logs Cisco:NIR:2.0 <<password>>
```

- To collect the logs for applications such as Kafka and Elastic execute the following command.

```
# appmgr afw fetch-logs Cisco:Kafka
```

- To collect the logs for all applications execute the following command.

```
# appmgr techsupport
```

