Note

Before using this information and the product it supports, read the information in “Notices” on page 51.

This edition applies to version 1.2.2.0 of IBM Network Performance Insight and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 2015, 2018.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Introduction ........................................... v
  Intended audience .................................. v
  Service Management Connect. .................. v
  Network Performance Insight technical training ... v
  Support information ................................ v
  Conventions used in this publication .......... vi
    Typeface conventions .............................. vi

Chapter 1. Network Performance Insight
value propositions ........................................... 1

Chapter 2. Network Performance Insight
architecture .................................................. 5
  Foundation services ................................ 8
    Manager ........................................... 9
    DNS ............................................. 9
    Event .......................................... 9
    Storage ........................................ 10
    UI ............................................. 14
  Entity Metric services .............................. 15
  Tivoli Network Manager Collector .................. 15
  Formula Service ..................................... 17
  SNMP Collector .................................... 17
  Entity Analytics ................................. 20
  Threshold ....................................... 20
  Flow Metric services .............................. 22
    Flow Collector ................................ 23
    Flow Analytics ................................ 37
  Remote Flow Collector ............................ 41
  Performance Metric OOTB Device Support ...... 41
  Network Performance Insight Dashboards .... 43

Chapter 3. Deployment considerations ............. 45
  Suggested node and services layout ............ 45
    Cluster behavior ................................ 47

Notices ................................................. 51
  Trademarks ........................................ 53
  Terms and conditions for product documentation 54
Introduction

IBM® Network Performance Insight is a network performance monitoring system. It provides a comprehensive and scalable visibility on network traffic with visualization and reporting of network performance data for complex, multivendor, multi-technology networks.

The Network Performance Insight overview provides a description of the product, its features, and the steps to take if you decide to purchase IBM Network Performance Insight, Version 1.2.2.

Intended audience

The audience includes those considering IBM Network Performance Insight as a network performance monitoring solution and all the new users who want an overview of the system.

Service Management Connect

Connect, learn, and share with Service Management professionals and product support technical experts who provide their perspectives and expertise.

Access Network and Service Assurance community at https://www.ibm.com/developerworks/servicemanagement/nsa/index.html Use Service Management Connect in the following ways:

• Become involved with transparent development, an ongoing, open engagement between other users and IBM developers of Tivoli products. You can access early designs, sprint demonstrations, product roadmaps, and prerelease code.
• Connect one-on-one with the experts to collaborate and network about Tivoli and the Network and Service Assurance community.
• Read blogs to benefit from the expertise and experience of others.
• Use wikis and forums to collaborate with the broader user community.

Related information:

IBM Network Performance Insight community on developerWorks

Network Performance Insight technical training

For Tivoli technical training information, see the following Network Performance Insight Training website at https://tnpmsupport.persistentsys.com/updated_trainings

Support information

If you have a problem with your IBM Software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

Online


IBM Support Assistant

The IBM Support Assistant is a free local software serviceability workbench
that helps you resolve questions and problems with IBM Software products. The Support Assistant provides quick access to support-related information and serviceability tools for problem determination. To install the Support Assistant software, go to https://www.ibm.com/software/support/isa

Troubleshooting Guide
For more information about resolving problems, see the problem determination information for this product.

Conventions used in this publication
Several conventions are used in this publication for special terms, actions, commands, and paths that are dependent on your operating system.

Typeface conventions
This publication uses the following typeface conventions:

**Bold**
- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as Tip:, and Operating system considerations:)
- Keywords and parameters in text

*Italic*
- Citations (examples: titles of publications, diskettes, and CDs)
- Words defined in text (example: a nonswitched line is called a point-to-point line)
- Emphasis of words and letters (words as words example: "Use the word that to introduce a restrictive clause."; letters as letters example: "The LUN address must start with the letter L.")
- New terms in text (except in a definition list): a view is a frame in a workspace that contains data.
- Variables and values you must provide: ... where myname represents....

**Monospace**
- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

**Bold monospace**
- Command names, and names of macros and utilities that you can type as commands
- Environment variable names in text
- Keywords
- Parameter names in text: API structure parameters, command parameters and arguments, and configuration parameters
• Process names
• Registry variable names in text
• Script names
Chapter 1. Network Performance Insight value propositions

Network Performance Insight is a performance monitoring and diagnostic solution that is integrated with IBM Netcool® Operations Insight. It can help enterprises detect, isolate, and diagnose anomalous network events and maintain optimal performance of IT resources.

Currently, Network Performance Insight supports Flexible NetFlow. Flexible NetFlow allows a granular and accurate traffic measurements and high-level aggregated traffic collection.

It is built on state-of-the-art microservices that are configured and managed on Ambari. It also has the Hadoop-based big data architecture for elastic scalability and performance. For more information, see Chapter 2, “Network Performance Insight architecture,” on page 5.

You can derive the following benefits from Network Performance Insight:

**Proactive WAN monitoring**

Network Performance Insight can test for network issues such as latency and packet loss. Identifies and solves problems before the impact gets noticeable. Understands the network behavior patterns and can help in the following ways:

- Prevent network outages and slow downs.
- Detect policy violations and required changes to configuration settings.
- Monitor congestion with QoS queuing hierarchy.
- Generate alerts based on user-configured thresholds before an incident occurs.

Provide predefined and custom threshold settings that help provide alerts on traffic data violations at interface level. You can analyze and correlate this information from highly customizable Network Performance Insight Dashboards.

- Define and manage IP Grouping for usage-based accounting and billing.
- Plan the IP routing and peering agreements with the help of Autonomous systems.
- Make informed decisions based on historical data.
- Improve enterprise network management to optimize costs and increase organizational productivity.

**Micro and macro level management of enterprise networks**

Keeps performance, availability, and productivity of networks, applications, and devices as the key focus that results in good user-experience. Network Performance Insight provides these specific monitoring capabilities:

**Network performance monitoring**

Correlates interfaces with QoS Queue Drops metric value and traffic breakdown by Traffic class. Validates the effectiveness of network QoS policy.

Ensures that WAN capacity is maintained for business-critical applications with policy-based access control.
Reports IP SLA operation results in real-time to indicate general user experience degradation and helps the Operations Engineers to understand the impact of network traffic load on overall quality of latency sensitive network services.

Gives insight on the impact of network traffic load on business critical and business irrelevant applications.

**Application health monitoring**

With the help of advanced Cisco NBAR2 (Next Generation Network-Based Application Recognition), it can detect the applications that consume high network bandwidth.

Identifies the applications in your network without relying only on the port numbers and detects applications latency based on the Application Response Time and Total Application Delay. Control access to non-business relevant applications that consume network bandwidth.

**Device health monitoring**

SNMP data is supplied as ready-to-use performance packages for Cisco, Juniper, and Huawei devices that is supplied from Performance Metric OOTB Device Support. A single discovery and SNMP polling stack is used to look up and store the data from Tivoli® Network Manager to Network Performance Insight.

**At-a-glance visibility into network performance**

With exhaustive network visibility, Network Performance Insight helps you identify the highest and least used assets (network, infrastructure, and applications). Navigate from high-level dashboard views to drill-down and pivot to pinpoint the issues from low-level details. It has options for scalable data collection and real-time visibility of data. Built-in visualizations in Network Performance Insight are as follows:

- Network Performance Overview
- NetFlow dashboards
- On Demand Filtering dashboards

These dashboards and widgets can provide the following capabilities to Network Operations Engineers, Global Operations Managers, and Network Capacity Planners:

- Operational insights
- Analytical insights
- Strategic insights
- On-demand filtering insights

**Fast-tracked troubleshooting and incident management**

Improves visibility into network performance to help minimize service degradations and disruptions, and speed up troubleshooting. The usage and trend analysis of the reports remove network bottlenecks, blind spots with seamless navigation among the dashboards and performance monitoring protocols to allow immediate problem resolution.

The clear data points in the reports help to differentiate bandwidth spikes, along with information about congestion in network. Reduced MTTR (Mean Time To Resolution) can improve the ROI of an enterprise.
Root cause analysis and capacity planning

Root cause analysis can help in strategic planning and not only solve the current network issues but also plan for the future expansions and upgrades. Highlights trends in network traffic so that you can plan bandwidth capacity needs. Analyze traffic planning with capacity planning reports.

Partial support for security incident identification

Heightened network bandwidth, unidentifiable devices, and interfaces can lead to suspicion of a security threat to your network. You can correlate a possible security threat with a set of specific KPIs from Network Performance Insight visualizations.

Related information:

- [Flexible NetFlow](#)
- [Cisco AVC Solution Overview](#)
IBM Network Performance Insight is a network performance monitoring system. It offers both real-time and historical trends in network performance and interactive view on the network data that helps in reduced network downtime and optimized network performance.

Network Performance Insight provides IBM Netcool Operations Insight with comprehensive IP network device performance monitoring and session traffic analysis.

The following diagram shows how data is flowing through the various components in Network Performance Insight:

**Network Performance Insight services**

Network Performance Insight services are running on microservice architecture that has the software application as a suite of independently deployable, small, modular services in which each service runs a unique process and communicates through a well-defined, lightweight mechanism. Currently, Network Performance Insight 1.2.2 consists of the following microservices:

© Copyright IBM Corp. 2015, 2018
Foundation services
- Manager
- DNS
- Event
- Storage
- UI

Entity Metric services
- Tivoli Network Manager Collector
- SNMP Collector
- Formula Service
- Entity Analytics
- Thresholds

Flow Metric services
- Flow Collector
- Flow Analytics

For more information about these services, see their respective sections in *IBM Network Performance Insight: Product Overview*.

Network Performance Insight additional components

Some of the additional components that are introduced in Network Performance Insight V1.2.2 for enhanced functions are described here:

**Performance Metric OOTB Device Support**

Performance Metric OOTB Device Support provides some additional pre-loaded, vendor-specific device performance SNMP metrics that can be discovered and polled from Tivoli Network Manager system and the data can be displayed on Network Performance Insight dashboards.

For more information, see “Performance Metric OOTB Device Support” on page 41.

**Network Performance Insight Dashboards**

These interactive dashboards are the built-in JSON-based dashboards suite that can display aggregated network data from Network Performance Insight database with the help of REST API calls. It supports a combination of data from multiple data sources.

This feature provides a wide variety of dashboards for Network Operators, Network Engineers, and Network Capacity Planners. These dashboards help in pinpointing the troubled resources and general resource performance. A number of web-based configuration options are available to control the data that is displayed on the dashboards.

For more information, see *Network Performance Insight Dashboards* section in *IBM Network Performance Insight: Product Overview*.

**Note:** Networks for Operations Insight is a solution extension of Netcool Operations Insight that includes the following components and products:
- Tivoli Network Manager
- Tivoli Netcool Configuration Manager
IBM Open Platform with Apache Spark and Apache Hadoop components

IBM Open Platform with Apache Spark and Apache Hadoop (IOP) can be used to help process and analyze the volume, variety, and velocity of data that continually enters your organization every day. Network Performance Insight is installed as a service extension to the installed IBM Open Platform with Apache Spark and Apache Hadoop stack.

The features of IOP that are used in Network Performance Insight:

- IBM Open Platform with Apache Spark and Apache Hadoop
- Default support for rolling upgrades for Hadoop services
- Support for long-running applications within YARN for enhanced reliability
- Spark in-memory distributed compute engine for dramatic performance increase
- Apache Ambari operational framework. Apache Ambari is an open framework for provisioning, managing, and monitoring Apache Hadoop clusters. Ambari provides an intuitive and easy-to-use Hadoop management web UI backed by its collection of tools and APIs that simplify the operation of Hadoop clusters.
- Essentially includes the following open source technologies for working with Network Performance Insight:
  - HDFS
  - Kafka
  - Ambari
  - Spark
  - ZooKeeper

Note: Because Zookeeper requires a majority, it is best to use an odd number of machines. For example, with four machines ZooKeeper can handle the failure of a single machine; if two machines fail, the remaining two machines do not constitute a majority. However, with five machines ZooKeeper can handle the failure of two machines.

Integrated products

Products that are integrated with Network Performance Insight 1.2.2:

**Jazz™ for Service Management**

Dashboard Application Services Hub provides visualization and dashboard services in Jazz for Service Management. It has a single console for administering IBM products and related applications. Visualization for Network Performance Insight is federated into Dashboard Application Services Hub.

**IBM Tivoli Network Manager IP Edition**

Tivoli Network Manager provides network discovery, device polling, including storage of polled SNMP data for reporting and analysis, and topology visualization. In addition, Network Manager can display network...
events, perform root-cause analysis of network events, and enrich network events with topology and other network data.

**Tivoli Netcool/OMNibus component of IBM Netcool Operations Insight**

Netcool Operations Insight is powered by the fault management capabilities of IBM Tivoli Netcool/OMNibus. In Network Performance Insight V1.2.2, Tivoli Netcool/OMNibus is an important part of the solution for monitoring the network threshold violations.

**Related information:**
- IBM Network Performance Insight on IBM Knowledge Center
- IBM BigInsights 4.2 documentation
- HDFS Architecture
- Apache Hadoop YARN
- Apache Kafka
- Apache Zookeeper
- IBM Networks for Operations Insight

**Foundation services**

Foundation services are the basic infrastructure services that are used by multiple other Network Performance Insight services.

- **Manager**
  - Monitors the status and health of all Network Performance Insight microservices.

- **DNS**
  - Operates as a cluster singleton. The DNS service is stateless. Its performance depends on the performance of the supporting DNS server that is provided by your enterprise.

- **Event**
  - Operates as a cluster singleton. The performance of this service depends on the performance of the STDIN probe and Tivoli Netcool/OMNibus server. STDIN probe is used to send events to the IBM Tivoli Netcool/OMNibus server. The terms alert and event can be used interchangeably.

- **Storage**
  - Operates as a cluster singleton that provides resiliency. IBM Network Performance Insight uses a built-in columnar storage database.

- **UI**
  - Operates in cluster load balancing mode. Each instance of UI is fully operational and capable of serving requests.
Manager

Read this information for some salient features of the Manager service.

- Periodically monitors the status of all Network Performance Insight services.
- Sends metrics to Ambari.
- Provides the `basecamp-manager-cmd` command line interface to start, stop, and check the status of selected Network Performance Insight microservices from Ambari. The `basecamp-manager-cmd` command script is available in `/opt/IBM/basecamp/basecamp-manager/bin` folder.

  For more information about the command, see `basecamp-manager-cmd` command reference in *IBM Network Performance Insight: References*.
- Copies the Spark and Hadoop configuration files to HDFS for the first time after the installation is complete.

DNS

The DNS Service resolves DNS names for reporting and distributing the interface metadata.

By default, the network requests that support DNS lookup run on port 53 for TCP and UDP protocols. The configuration settings are available on Ambari server from Services > NPI > NPI Settings under NPI DNS Services pane.

Event

Read this information to know how Event Service interacts with other services to handle event generation.

"Flow Analytics” on page 37 Service processes and manages events for Flow data and "Threshold” on page 20 Service for Entity Metric data. The Flow Analytics Service and Threshold Service generate alerts when any violation occurs.

Event evaluation process

Thresholds are computed based on the data that is collected from 1-minute aggregation.

The event evaluation process goes through these phases:

1. The Event Service then compares for any threshold violations.
   - If a KPI enters a violation state of [Major and critical severity] an event is generated. The Event Viewer displays the active events from IBM Netcool Operations Insight dashboards.

2. If an existing threshold violation state changes, an event is generated.
   - The severity is set to CLEAR and the event from the Event Viewer is removed. Whether a cleared event is removed from Tivoli Netcool/OMNIbus Event Viewer or not is determined by Tivoli Netcool/OMNIbus Web GUI configurations.

For more information about exiting a threshold violation in Network Performance Insight, see "Exiting a threshold violation"

Generating an event

An event is generated only in the following conditions:
- KPI value violates the threshold configuration for an interface and no existing violation for the same interface is triggered.
- The threshold state is cleared.

The two main event management scenarios:

**Generating a new event**
When the following conditions are met, an event is generated and sent to IBM Netcool Operations Insight dashboards:
- Traffic data violates the threshold configurations.
- The severity is not changed for a KPI that has violated a threshold.

**Note:** If the severity is not change for of a KPI that is already in a violated state, a new event is not generated.

**Generating an event when the severity changes**
When the following conditions are met, an event is generated and sent to IBM Netcool Operations Insight dashboards:
- The Traffic data that is in violated state changes to a new severity.

**Storage**

Network performance data profiling requires advanced database technology to handle high volumes of data in a large enterprise and service provider infrastructures. Network Performance Insight uses an in-built database with efficient memory usage.

It has the following features:
- Compressed, columnar data storage that manages the data that is collected by Network Performance Insight.
- Supports high-bandwidth queries and analysis of data.
- Provides inbound APIs to insert and update data.
- Maintains HDFS file-store, aging data, query coordination.
- Periodically consolidates storage files and purging old data with the help of Storage optimizer.
- Queries are delegated to spark for scalability. All fact data are stored as parquet files in compressed, columnar format which can be processed by spark efficiently.
- Provides a mechanism for incremental storage of network performance data for specific time intervals.

The Network Performance Insight Storage Service uses Hadoop-based big data infrastructure for its distributed computing features.

Hadoop and Spark are a fully integrated infrastructure solution with cluster management and analytics software that is optimized for Hadoop and Spark based workloads. Hadoop comprises of two main components; the HDFS, and a Java™ based file system for storing large volumes of data and a programming paradigm, that is, Hadoop MapReduce.

The Storage Service state is maintained by the following components:
- HDFS - fact tables
- Kafka - domain tables
- Application logic - database schema
• Table Schema information - Kafka Schema Registry

State is automatically recovered at start up. Scaleout is achieved by using Spark to support queries.

**Storage optimization**

Network Performance Insight database is optimized to improve Network Performance Insight overall performance. Storage optimization minimizes the data retrieval time and reduces hardware and administration costs.

**Data storage**

Database schema that contains various tables to store specific types of data that is collected by Network Performance Insight.

The following table lists the types of data that is stored in Network Performance Insight database schema:

*Table 1. Network Performance Insight database schema.*

<table>
<thead>
<tr>
<th>Database schema</th>
<th>Type of data</th>
</tr>
</thead>
</table>
| Flow Metric     | The following Flow metric data is stored in this schema:  
|                 | • Flow Raw data  
|                 | • Aggregated data of 1 minute, 30 minutes, and daily for all top 10 performers  
|                 | • Aggregation status  
|                 | • Interface configuration settings for Flow metric data  
|                 | • The target information for Application response time is stored in this table.  
|                 | • All the IP addresses of the various Flow-enabled devices are stored in this table.  
|                 | • The following QoS information:  
|                 |   – QoS class names, class IDs, and class types  
|                 |   – QoS policy names, policy IDs, and policy types  
|                 |   – QoS queue IDs for the different entities  
| Configuration   | The following Flow configuration data is stored in this schema:  
|                 | • Aggregation  
|                 | • Autonomous systems  
|                 | • Domain Names  
|                 | • IP Address Group  
|                 | • NBAR  
|                 | • NBAR2  
|                 | • Retention Profiles  
|                 | • Threshold definitions  
|                 | • ToS data  
| Event           | Event data for both Flow and entity metrics data is stored in this schema.  

### Table 1. Network Performance Insight database schema (continued).

<table>
<thead>
<tr>
<th>Database schema</th>
<th>Type of data</th>
</tr>
</thead>
</table>
| Inventory       | Inventory-related metadata that is collected from IPSLA enabled devices is stored in this schema:  
  - IP SLA enabled device properties names and values.  
  - IP SLA enabled device formulas or metric names.  
  - IP SLA enabled devices entity or resource names and parent entity names.  
  - IP to host name mappings.  
  - Network interface information that is discovered by Tivoli Network Manager and available in Network Connectivity and Inventory Model (NCIM) database. |
| Defaults        | Dashboard Application Services Hub integration-related data is stored in this schema. |
| NCIM            | The following type of data is stored in this schema:  
  - Discovered network interface related data that corresponds to related devices.  
  - Data that is polled from Tivoli Network Manager database by using Kafka Connect, which is sent to Kafka message bus and stored within Network Performance Insight database schema. |
| NCPOLLDATA      | Resources and metric metadata from the following sources is stored in this schema:  
  - The Polling engine, ncp_poller from Tivoli Network Manager uses data from the tables in this schema to determine how to poll. A copy of the same data from Tivoli Network Manager is stored in this schema.  
  - Tivoli Network Manager core server contains Apache Storm server that manages poll data aggregation. The data is pushed to Kafka message bus and stored within Network Performance Insight database schema. |
| Threshold       | The following Threshold-related data for entity metrics is stored in this schema:  
  - Threshold evaluation records and static threshold definitions of entity metric data.  
  - Static threshold definitions for entity metrics data. |
| Entity Metric   | The following Entity metric data is stored in this schema:  
  - Entity Metric raw data  
  - Tivoli Network Manager metric data and SNMP data from IP SLA enabled devices.  
  - Entity Metric aggregated data of 30 minutes, 6 hours and daily  
  - Entity Metric aggregation status |

### Flow Aggregated data

Network Performance Insight consists of Fact tables and Domain Objects.

All raw NetFlow data is stored up to 5 days as the initial default configuration of Network Performance Insight in the Flow database table.
The data for all interfaces from the NetFlow Top Talkers on Traffic Details dashboard is aggregated with every 1-minute interval. Aggregated data is progressively stored in 1 minute, 30 minutes, and Daily aggregation tables. The most recent data is available with 1-minute aggregation. Data less than 50 hours are populated from 1-minute aggregation tables while data more than 50 hours but less than 100 days are populated from 30-minutes aggregation tables. All data that is more than 100 days are populated from 1-day aggregation tables.

The following table lists the traffic tables that are queried for the data that is displayed in dashboards in Network Performance Insight for different time intervals:

<table>
<thead>
<tr>
<th>Time interval</th>
<th>Traffic tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 50 hours</td>
<td>1-minute aggregation</td>
</tr>
<tr>
<td>50 hours to less than 100 days</td>
<td>30-minutes aggregation</td>
</tr>
<tr>
<td>More than 100 days</td>
<td>1-day aggregation</td>
</tr>
</tbody>
</table>

**Entity Metric Aggregated data**

All Tivoli Network Manager SNMP raw data is stored up to 10 days as the initial default configuration of Network Performance Insight in the Entity Metric raw database table.

The most recent data is available with 30-minutes aggregation. Data is populated from the 30 minutes, 6-hours aggregation, or 1-day aggregation table based on the time period and aggregation type from the relevant reports.

**Related concepts:**

"Built-in aggregation definitions" on page 38

This section details the built-in aggregation types and their grouping keys. Based on these aggregations the data for the Top Talker views from the Traffic Details dashboard is populated. These aggregations are user configurable.

**Retention period**

Network Performance Insight supports data retention, which relates to historical data and managing the data storage periods.

You can maintain Network Performance Insight database by configuring the retention period to keep the required data. The time period for which you can store raw, aggregated, resolved data by DNS, and events types of data, depends on the number of flows that are received by Network Performance Insight and the free disk space available on your system.

You can maintain Network Performance Insight database by configuring the retention period to keep the required data.

After the specified period, the database is cleaned up automatically. For resolved data by DNS, after the specified retention period, DNS server auto resolves the hostname. If the DNS resolution is unsuccessful, the IP address is populated instead.

The following table lists the default retention period that is used for the following types of data:
<table>
<thead>
<tr>
<th>Type of data</th>
<th>Retention period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow Raw</td>
<td>5 Days</td>
</tr>
<tr>
<td>Flow 1 minute aggregated</td>
<td>1 Month</td>
</tr>
<tr>
<td>Flow 30 minutes aggregated</td>
<td>12 Months</td>
</tr>
<tr>
<td>Flow Daily aggregated</td>
<td>12 Months</td>
</tr>
<tr>
<td>DNS</td>
<td>3 Months</td>
</tr>
<tr>
<td>Events</td>
<td>6 Weeks</td>
</tr>
<tr>
<td>Logs</td>
<td>10 Days</td>
</tr>
<tr>
<td>Entity Raw</td>
<td>10 Days</td>
</tr>
<tr>
<td>Entity 30 minutes aggregated</td>
<td>30 Days</td>
</tr>
<tr>
<td>Entity 6 hours aggregated</td>
<td>30 Days</td>
</tr>
<tr>
<td>Entity Daily aggregated</td>
<td>30 Days</td>
</tr>
<tr>
<td>Entity Threshold State</td>
<td>90 Days</td>
</tr>
</tbody>
</table>

For example, if you consider the date as 12 January, the data is retained from 12 January until 12 April before it gets purged for Entity Raw.

**Note:**
- The specified disk space usage is calculated based on the default *Hardware requirements*.
- All these options except the logs retention period can be configured from the Network Performance Insight System Configuration UI.
  - To configure your own settings for database retention period for logs, see *Configuring logging in Troubleshooting IBM Network Performance Insight*.
  - To configure the rest of the database retention period profiles, see *Administering retention profiles in Administering IBM Network Performance Insight*.

**UI**

The UI Service controls all the visualizations that are associated with Network Performance Insight.

Jazz for Service Management federation can be configured in one of the following ways:
- A single instance of Jazz for Service Management can be configured to federate against any one of the UI instances. If either Jazz for Service Management server or the UI instance that it is federating fails, all UI functions are disabled.
- A HTTP/HTTPS load balancer can be deployed between a single instance of Jazz for Service Management and a cluster of UI instances. The load balancer can be configured to present a single IP address for the pool of UI instances. Load is distributed by the load balancer and if one or more UI instances fail, the others pick up the load.
- The UI Service also contains all the files that are related to Network Performance Insight Dashboards. These dashboards are technology neutral-JSON files and require properties files.
  H2 database is used to store the core database tables. The H2 database stores the dashboards data. Some of the information that is stored in H2 database; the list of dashboards, users, roles, page-role mapping information, dashboard state, database queries that are used for dashboard pages.
Network Performance Insight Dashboards application is deployed on WebSphere Application Server that is available on Dashboard Application Services Hub.

The configuration settings are available on Ambari server from Services > NPI > NPI Settings under NPI Web Services pane.

Dashboard Application Services Hub federation settings are available on Ambari server from NPI > NPI Core Settings under Advanced > Advanced npi-auth pane.

Entity Metric services

Services that are required for Network Performance Insight entity metric data that is collected, aggregated, and monitored.

- **“Tivoli Network Manager Collector”**
  Operates as a cluster singleton. The Tivoli Network Manager Collector Service periodically polls IP SLA probes that are discovered by Tivoli Network Manager and perform reconciliation.

- **“Formula Service” on page 17**
  Operates as a cluster singleton. The Formula Service calculates metric values for the data that is collected by SNMP Collector. It uses formulas that are deployed against specific entity types.

- **“SNMP Collector” on page 17**
  Operates as a cluster singleton. The SNMP Collector Service can poll SNMP enabled devices to provide metric polling data to Network Performance Insight. The SNMP Collector Service receives polling requests from Kafka, dispatches the appropriate SNMP requests, and places the resulting metric data back to Kafka for further analytic operations.

- **“Entity Analytics” on page 20**
  Entity Analytics Service is used to perform aggregation of RAW data that is collected from Tivoli Network Manager.

- **“Threshold” on page 20**
  Supports burst or static thresholds. Network Performance Insight and Tivoli Network Manager perform threshold evaluations on the entity metric data by using period-based data aggregations against the threshold definitions.

Tivoli Network Manager Collector

Tivoli Network Manager uses the Apache Storm real-time computation system to aggregate raw poll data into historical poll data, and stores raw and historical poll data in the NCPOLLDATA database. The Storm Spout provides polling data and related metadata through Kafka. Periodically, Spout checks the NCPOLLDATA database for new metric data and new collection metadata.

The Tivoli Network Manager Collector Service periodically polls IP SLA probes that are discovered by Tivoli Network Manager and perform reconciliation. If the new probes are discovered, a new request is generated and if a probe is stopped, the existing request is withdrawn. It pulls the SNMP metrics that are polled and IP SLA probes that are discovered from Tivoli Network Manager to Network Performance Insight Storage Service.

Metric data from Tivoli Network Manager is handled as follows:

- Network Performance Insight receives data from three Tivoli Network Manager NCPOLLDATA tables:
- **POLLDATA**
  Contains the actual Tivoli Network Manager polled data that includes references to the instance and object, the timestamp, and metric value.

- **MONITOREDOBJECT**
  Identifies the Tivoli Network Manager metrics that are being computed or polled.

- **MONITOREDINSTANCE**
  Identifies the Tivoli Network Manager interfaces or entities that are being polled.

Storm Spout in Tivoli Network Manager supports requests for sending full sets of MONITOREDOBJECT (metrics) and MONITOREDINSTANCE (resources) data. These requests are initiated by messages on the data request topic. Storm Spout pushes the data that is available in Tivoli Network Manager MONITOREDOBJECT and MONITOREDINSTANCE tables and populates the data into the same tables in Network Performance Insight.

See Configuring Apache Storm Spout in Network Manager in Installing and Configuring IBM Network Performance Insight.

- Kafka Connect polls the Tivoli Network Manager database by using JDBC and populates the NCIM.NETWORK_INTERFACE table in Network Performance Insight database.

  The Tivoli Network Manager Collector Service refers to Network Performance Insight configuration settings to connect to Tivoli Network Manager database.

  For more information about these settings, see Configuring communication with Tivoli Network Manager in Installing and Configuring IBM Network Performance Insight.

  The configuration settings are available on Ambari server from Services > NPI > Configs > NOI Core Settings under NOI Components > NOI SNMP Collector pane. These settings are populated with default information.

The Tivoli Network Manager Collector Service has the Kafka Consumers for each inbound data type. The Kafka Consumers receive the Tivoli Network Manager data from the Kafka topic and publish it on an internal event stream.

The collector service collates the Tivoli Network Manager data and writes the records into npi.timeseries.raw kafka topic. The data from the Kafka topic is picked up by the Storage Service and written to ENTITY_METRIC.RAW table.

The Tivoli Network Manager poll data is made available to other Network Performance Insight services through the Storage Service.

**Related information:**

- [Tivoli Network Manager](#)
- [Enabling the integration with Network Performance Insight](#)

**Data flow**

Use this information to understand how the IP SLA metric data is transferred and processed from Tivoli Network Manager system and other services in Network Performance Insight system.
Formula Service

Based on the list of IP SLA probes that are detected by Tivoli Network Manager Collector, the Formula Service creates poll definition requests for SNMP Collector to start the polling. Formula Service detects new IP SLA Probes or deactivates probes and sets up SNMP collector for metric collection.

Currently, the Formula Service supports SNMP data from Cisco IPSLA devices only. The Formula Service requests SNMP data on probes and uses the same polling interval that the probes use to measure RTT. If a probe is configured to measure every 30 seconds, then the Formula Service requests OIDs on the probe instance every 30 seconds.

The Formula Service obtains data to be used in formula execution and metric calculation from the npi.snmp.poll.data Kafka topic. All the calculated metric values are submitted to npi.timeseries.raw Kafka topic. This data is picked up by the Storage Service and written to ENTITY_METRIC.RAW table.

For more information about all the supported Cisco IP SLA formulas, see SNMP formulas section in IBM Network Performance Insight: References

For more information about the Performance Metric OOTB Device Support formulas, see SNMP formulas section in IBM Network Performance Insight: References.

SNMP Collector

SNMP enabled devices can be configured to probe and measure how traffic is flowing across the network with the help of Cisco IP SLA metric data such as response times, latency, jitter, and packet loss. This information can be used to determine the current performance of the network from the user perspective.

SNMP Collector Service collects Cisco IP SLA-specific data and provides SNMP IP SLA network monitoring for specific quality of service measurements. The SNMP Collector receives poll requests and SNMP credentials from Kafka topics.
Polling definitions

By default, the SNMP Collector receives polling definitions from a Kafka topic by name npi.snmp.polldefinitions. The polling definition messages are in the following format:

[resource]/[interval]/[attribute]

Where:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource</td>
<td>The resource that must be polled.</td>
</tr>
<tr>
<td>attribute</td>
<td>The OID on the resource that must be polled.</td>
</tr>
<tr>
<td>interval</td>
<td>The time interval that the OID or instance on the resource that must be polled.</td>
</tr>
</tbody>
</table>

For example:

```
{
    "resource":"10.55.239.31:161",
    "interval":900000,
    "attribute":"1.3.6.1.4.1.9.9.42.1.2.10.1.2.100"
}
```

Polling credentials

By default, the SNMP Collector receives SNMP credentials from a Kafka topic by name npi.snmp.poll.credentials. The polling definition messages are in the following format:

```
{
    "agent":{
        "retries":2,
        "version":1,
        "port":161,
        "ipAddress":"10.55.239.4",
        "timeout":10000
    },
    "credentials":{
        "readCommunity":"j5pxixeQv/kv6ZHPYdSj3w==",
        "writeCommunity":"m7J4u0Y3d2cOTJqis8Fg+Q=="
    }
}
```

Where:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>Device</td>
</tr>
<tr>
<td>retries</td>
<td>How many times the SNMP helper and polling operations attempted to access a device.</td>
</tr>
<tr>
<td>version</td>
<td>SNMP version</td>
</tr>
<tr>
<td>port</td>
<td>UDP port number for IP SLA agent. By default, 161.</td>
</tr>
<tr>
<td>ipAddress</td>
<td>IP address of the resource that the credentials must be used.</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>timeout</td>
<td>The time in milliseconds to wait for a reply before timeout.</td>
</tr>
<tr>
<td>credentials</td>
<td></td>
</tr>
<tr>
<td>readCommunity</td>
<td>Encrypted read community string.</td>
</tr>
<tr>
<td>writeCommunity</td>
<td>Encrypted write community string.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The read and write community string protects the device against</td>
</tr>
<tr>
<td></td>
<td>unauthorized changes.</td>
</tr>
</tbody>
</table>

For SNMPv3, the message format is as follows:

```json
{
    "agent":{
        "retries":5,
        "version":3,
        "port":161,
        "ipAddress":"10.55.239.202",
        "timeout":10000
    },
    "credentials":{
        "username":"user3",
        "authPass":"60KNCad9c/IAo8rozinIzoELBJvVZK7",
        "privPass":"gQwwCNVvkrs=",
        "authType":"SHA1",
        "securityLevel":"authNoPriv",
        "privType":"nopriv",
        "defaultContext":"
    }
}
```

**Polling data**

By default, the SNMP collector produces polled data on a Kafka topic by name npi.snmp.poll.data. The polled data messages are in the following format:

`<resource>,<attribute>,<timestamp>,<value>`

For example:

```json
{
    "resource":"10.55.239.211:161",
    "attribute":"1.3.6.1.4.1.9.9.42.1.5.2.1.14.500",
    "timestamp":1495677950905,
    "value":"65"
}
```

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource</td>
<td>The resource that is polled. It must be in this format:</td>
</tr>
<tr>
<td></td>
<td>&lt;IP_address&gt;:&lt;port&gt;</td>
</tr>
<tr>
<td>attribute</td>
<td>The OID on the resource that is polled. It must be in this format:</td>
</tr>
<tr>
<td></td>
<td>&lt;oid&gt;,&lt;probe_ID&gt;</td>
</tr>
<tr>
<td>timestamp</td>
<td>The time stamp that the OID or instance on the resource that is polled.</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>value</td>
<td>The value that is retrieved from the OID or instance on the resource.</td>
</tr>
</tbody>
</table>

**Entity Analytics**

The Entity Analytics Service performs 30 minutes, 6 hours and 1-day aggregations.

The Tivoli Network Manager collector writes entity metric RAW records in parquet file format on HDFS every minute and sends an aggregation status message to indicate that a new RAW data is available. It checks and triggers the next aggregation when the data is available.

It refines the raw data, filters the results, and aggregates the KPI values. The values are aggregated by sum, min, max, and count. The results are then stored in Network Performance Insight database.

The Entity Analytics Service operates as a cluster singleton. It delegates queries to Apache Spark to achieve faster query response time.

**Threshold**

A threshold is a value that is compared against the predefined threshold configurations. It is evaluated to see whether it violates a specific restriction. The primary objective of thresholding is to determine any violations and to generate alerts. When the value falls outside the acceptable threshold range, the system generates and stores the event condition and forwards it to the Event Management System.

**Static thresholds**

Static (Burst) thresholding is user-defined static values at specific intervals, which analyze data and generate events when a violation occurs.

If your IBM Netcool Operations Insight solution is integrated with Network Performance Insight, then you can define static thresholds for anomaly detection.

You can define a static threshold for a KPI within the poll definition that polls for that KPI. If these static thresholds are violated for any performance measure on a device or interface, IBM Tivoli Netcool/OMNibus events are generated at an appropriate severity level.

Thresholds define the status of an attribute based on specific conditions. You can enable threshold evaluation on a selected resource or interface. A threshold is violated when the result of the collected metric value is evaluated as exceeding (upper) or dropping (lower) to a specified configured threshold level. The actual evaluation and disposition depends on the threshold type, Upper, Lower, or Band.

To define an anomaly threshold, see [Defining anomaly thresholds](#).

Related information:

[Defining performance thresholds for anomaly detection](#)
**Threshold violation**

A metric is considered to be in a violation state when it breaches the limits of traffic volume data for a specific number of times according to the configuration settings.

**Threshold type**

The severity of a violation is decided based on the violation against either on the upper or lower limit that is configured for a threshold.

- If the threshold is configured to check for Over, then severity is determined as:
  - If KPI value is greater than or equal to the Upper Limit, then the severity is Critical.
  - If the KPI value is greater than or equal to the lower limit, but not exceeding the upper limit, then the severity is Major.
- If the threshold is configured to check for Under, then severity is determined as:
  - If the KPI value is lesser than or equal to the upper limit, but not dropping below the lower limit, then the severity is Major.
  - If KPI value is lesser than or equal to the lower limit, then the severity is Critical.
- If the threshold is configured to check for Band, then severity is determined as:
  - If the KPI value is greater than or equal to the upper limit, then the severity is Critical.
  - If KPI value is lesser than or equal to the lower limit, then the severity is Critical.

For example, if the limit type is Over, and the traffic volume data is greater than the upper limit, the severity of the threshold is set to Critical. If the traffic volume data is greater than the lower limit, the threshold severity is set to Major.

The following table summarizes how a threshold severity is set to either Major or Critical for different limit types:

*Table 2. Threshold severity for limit types.*

<table>
<thead>
<tr>
<th>Limit Type</th>
<th>Scenario</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Over</td>
<td>Traffic volume data greater than the upper limit</td>
<td>Critical</td>
</tr>
<tr>
<td></td>
<td>Traffic volume data greater than the lower limit</td>
<td>Major</td>
</tr>
<tr>
<td>Under</td>
<td>Traffic volume data lesser than upper limit</td>
<td>Major</td>
</tr>
<tr>
<td></td>
<td>Traffic volume data lesser than the lower limit</td>
<td>Critical</td>
</tr>
<tr>
<td>Band</td>
<td>Traffic volume data greater than the upper limit</td>
<td>Critical</td>
</tr>
<tr>
<td></td>
<td>Traffic volume data lesser than the lower limit</td>
<td>Critical</td>
</tr>
</tbody>
</table>

**Note:** For limit type Band, the severity is always set to Critical for any threshold violation scenarios.
Exiting a threshold violation

Exiting from a violation state is the action of setting the severity of an event to CLEAR. The severity is cleared after a KPI goes into a non-violation state. The threshold at which an event is raised then is cleared.

Non-Violation -> Major Violation -> CLEAR -> Non-Violation
Or
Non-Violation -> Critical Violation -> CLEAR -> Non-Violation

Network Performance Insight supports system generated auto clear. The following table shows, when the system can set the severity to CLEAR.

*Table 3. Scenarios for CLEAR severity.*

<table>
<thead>
<tr>
<th>Limit Type</th>
<th>Current Severity</th>
<th>Scenario</th>
<th>New Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Over</td>
<td>KPI is either in Major or Critical level</td>
<td>Traffic volume data lesser than the lower limit</td>
<td>CLEAR</td>
</tr>
<tr>
<td>Under</td>
<td>KPI is either in Major or Critical level</td>
<td>Traffic volume data greater than upper limit</td>
<td>CLEAR</td>
</tr>
<tr>
<td>Band</td>
<td>KPI is in Critical level</td>
<td>Traffic volume data falls within the range of upper limit and lower limit</td>
<td>CLEAR</td>
</tr>
</tbody>
</table>

If an existing threshold configuration is modified, by changing any of the configuration values such as limit type, upper or lower limit, number of events or disable the threshold, the system sets the severity to CLEAR.

For more information about the configuration steps, see *Configuring Flow Thresholds* in *Using IBM Network Performance Insight.*

**Flow Metric services**

Services that are required for Network Performance Insight traffic flow data that is collected, aggregated, and monitored.

- "Flow Collector” on page 23
  
  Flow Collector Service is responsible for collecting IP traffic information from flow-enabled devices.

- "Flow Analytics” on page 37
  
  Flow Analytics Service performs aggregations on the Flow data to compute Top-N resource aggregations.

- "Remote Flow Collector” on page 41
  
  Remote Flow Collector is an extension of the Flow Collector for the collector to coexist with your exporters to prevent the loss of data collection. It operates as a cluster singleton.
Flow Collector

A Flow is a sequence of packets with common characteristics such as same source and destination IP address, transport layer port information, and type of protocol. The Flow enabled devices or exporters collect Flow data from the network.

The Flow Collector Service in Network Performance Insight performs these basic functions:

- Receives Flow records from Flow-exporters.
- Parses, validates, and normalizes the various Flow record formats into a common format.
- Enriches and filters Interfaces based on enable/disable flag set per network interface.
- Limits the number of interfaces that are enabled in Network Performance Insight.
- Stores the normalized and enriched Flow records in Apache Parquet files in HDFS.
- Notifies the Storage Service of the availability of Flow RAW data files via Kafka.

The configuration settings are available on Ambari server from Services > NPI > Configs > NPI Settings under NPI Components > NPI Flow Collector.

For more information about these configurations, see Configuring the Flow Collector Service in Installing and Configuring IBM Network Performance Insight.

Collection process

A collection process must be able to receive the flow information that is passing through multiple network elements within the data network.

Flow metric collection

Traffic on a data network can be perceived as a flow of data between two end-points that passes through network elements. For administrative or other purposes, it is beneficial to have the network elements observe these flows and report their characteristics to Network Performance Insight to understand the network usage patterns.

The three main components in NetFlow technology are; NetFlow cache, NetFlow exporter, and NetFlow collector.

NetFlow cache

A large amount of network information is condensed into a database of NetFlow information that is called the NetFlow cache. NetFlow can be configured to capture flows to the NetFlow cache. Typically, the NetFlow cache is constantly filling with flows and the router or switch searches the cache for flows that are terminated or expired and these flows are exported to the NetFlow collector.

NetFlow exporter

The NetFlow exporter sends flows that are in the cache to a NetFlow collector. NetFlow exporters are configured for Ingress interface traffic and Egress interface traffic or both.

NetFlow collector

NetFlow Collector receives and pre-processes flow data that is received from a flow exporter.
A flow is ready for export when it is inactive, when no new packets are received for the flow or if the flow is long lived (active) and lasts longer than the active timer. A flow is inactive if it has not received a packet for a specific duration that is longer than the inactive timeout value that is specified in the configuration. The flow record is deleted from the flow cache and an export record is generated, when the inactive timeout is triggered. By default, active timeout value is 30 minutes and inactive timeout value is 15 seconds.

The collector component in Network Performance Insight can then process and transform the data.

The collection process in Network Performance Insight is as follows:

**Collector requirements in Network Performance Insight**
Specifies the representation of different flows, the additional data that is required for flow interpretation, packet format, transport mechanisms that are used.

**Important:** For more information about flow concepts, see the specific vendor documents.
Flow version support

<table>
<thead>
<tr>
<th>Flow formats</th>
<th>Vendor</th>
<th>Supported versions</th>
<th>Supported traffic types</th>
<th>Supported protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetFlow</td>
<td>Cisco</td>
<td>1, 5, 9, and IPFIX</td>
<td>• IPv4</td>
<td>• UDP</td>
</tr>
<tr>
<td>JFlow</td>
<td>Juniper Networks</td>
<td>5, 9</td>
<td>• IPv6</td>
<td>• SCTP</td>
</tr>
<tr>
<td>CFLOW</td>
<td>Alcatel</td>
<td>5, 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NetStream</td>
<td>Huawei</td>
<td>5, 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPFIX</td>
<td>Industry standard</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: See, “RFC 7011”

For a complete list of IPFIX entities, see IP Flow Information Export (IPFIX)

NetFlow V9 formats:

V9 format is template-based. Templates provide an extensible design to the record format.

V9 packet layout

The NetFlow V9 record format consists of a packet header and at least one or more template or data FlowSets. A template FlowSet provides a description of the fields that will be present in future data FlowSets. These data FlowSets might occur later within the same export packet or in subsequent export packets.

V9 packet header format

<table>
<thead>
<tr>
<th>Packet Header</th>
<th>Template FlowSet</th>
<th>Data FlowSet</th>
<th>Data FlowSet</th>
</tr>
</thead>
</table>

V9 packet header fields

<table>
<thead>
<tr>
<th>Fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>The version of NetFlow records exported in this packet; for Version 9, this value is 0x0009</td>
</tr>
<tr>
<td>count</td>
<td>Number of FlowSet records (both template and data) contained within this packet</td>
</tr>
<tr>
<td>Fields</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SysUptime</td>
<td>Current time in milliseconds since the export device is started.</td>
</tr>
<tr>
<td>UNIX seconds</td>
<td>Current time in seconds that have elapsed since 00:00:00 Coordinated Universal Time, Thursday, 1 January 1970.</td>
</tr>
<tr>
<td>Sequence number</td>
<td>Incremental sequence counter of all export packets that are sent by this export device; this value is cumulative, and it can be used to identify any missed export packets. <strong>Note:</strong> This is a change from the NetFlow Version 5 and Version 8 headers, where this number represented “total flows.”</td>
</tr>
<tr>
<td>Source ID</td>
<td>The Source ID field is a 32-bit value that is used to guarantee uniqueness for all flows that are exported from a particular device. (The Source ID field is the equivalent of the engine type and engine ID fields that are found in the NetFlow Version 5 and Version 8 headers). The format of this field is vendor-specific. In the Cisco implementation, the first two bytes are reserved for future expansion, and is always zero. Byte 3 provides uniqueness about the routing engine on the exporting device. Byte 4 provides uniqueness about the particular line card or Versatile Interface processor on the exporting device. Collector devices must use the combination of the source IP address plus the Source ID field to associate an incoming NetFlow export packet with a unique instance of NetFlow on a particular device.</td>
</tr>
</tbody>
</table>

**Related information:**

- [NetFlow Version 9 Flow-Record Format](#)

**Template FlowSet format:**

One of the key elements in the new NetFlow V9 format is the template FlowSet.

**NetFlow V9 template FlowSet format**

Templates enhance the flexibility of the NetFlow record format because they allow a NetFlow collector or display application to process NetFlow data without necessarily knowing the format of the data in advance. Templates are used to describe the type and length of individual fields within a NetFlow data record that match a template ID.

**Related information:**

- [NetFlow Version 9 Flow-Record Format](#)

**Template FlowSet field descriptions:**

Template IDs are not consistent across a router restart. Template IDs must change only if the configuration of NetFlow on the export device changes.
## NetFlow V9 template FlowSet field descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlowSet ID</td>
<td>The FlowSet ID is used to distinguish template records from data records. A template record always has a FlowSet ID in the range of 0-255. Currently, the template record that describes flow fields has a FlowSet ID of zero and the template record that describes option fields has a FlowSet ID of 1. A data record always has a nonzero FlowSet ID greater than 255.</td>
</tr>
<tr>
<td>Length</td>
<td>Length refers to the total length of this FlowSet. Because an individual template FlowSet might contain multiple template IDs, the length value must be used to determine the position of the next FlowSet record, which might be either a template or a data FlowSet. Length is expressed in Type/Length/Value (TLV) format, meaning that the value includes the bytes used for the FlowSet ID and the length bytes themselves, and the combined lengths of all template records included in this FlowSet.</td>
</tr>
<tr>
<td>Template ID</td>
<td>As a router generates different template FlowSets to match the type of NetFlow data it is exporting, each template is given a unique ID. This uniqueness is local to the router that generated the template ID. Templates that define data record formats begin numbering at 256 since 0-255 are reserved for FlowSet IDs.</td>
</tr>
<tr>
<td>Field Count</td>
<td>This field gives the number of fields in this template record. Because a template FlowSet might contain multiple template records, this field allows the parser to determine the end of the current template record and the start of the next.</td>
</tr>
<tr>
<td>Field Type</td>
<td>This numeric value represents the type of the field. The possible values of the field type are vendor-specific. Cisco supplied values are consistent across all platforms that support NetFlow V9. At the time, of the initial release of the NetFlow V9 code (and after any subsequent changes that might add new field-type definitions), Cisco provides a file that defines the known field types and their lengths.</td>
</tr>
<tr>
<td>Field Length</td>
<td>This number gives the length of the defined field, in bytes.</td>
</tr>
</tbody>
</table>

Templates periodically expire if they are not refreshed. Templates can be refreshed in two ways:
• A template can be sent again every N number of export packets. Default template-resend interval of 20 packets, configurable 1-1000 data packets.

• A template can also be sent on a timer, so that it is refreshed every N number of minutes. Both options are user configurable. Default template-resend time of 10 minutes, configurable between 1 minute and 1 day.

Note: Both options are configurable by the user on the Exporter. When one of these expiry conditions is met, the Exporter must send the template FlowSet and Options template.

Related information:

NetFlow V 9 Flow-Record Format

Mediation process:

The Collector component receives template definitions from an exporter before the flow records are received. The Flow Records are then decoded and stored locally. If the template definitions are not received at the time a flow record is received, the Collector keeps the flow record for later decode after the template definitions are received.

The Collector does not assume that only one template FlowSet is present in an export packet. In rare circumstances, the export packet might contain several template FlowSets.

Templates live only for a certain time frame. The lifetime of a template is deducted on the Collector that is based on the time when the last template FlowSet is received from the exporter. The Collector does not attempt to decode the flow records with an expired template. The Collector maintains a similar list as follows:

<Exporter, Export Interface, Template ID, Template Def, Last Received>

If a new template definition is received when the exporter is restarted, it immediately overrides the existing definition.

Related information:

NetFlow Version 9 Flow-Record Format

IPFIX overview:

Internet Protocol Flow Information Export (IPFIX) is an IETF protocol, and the name of the working group that defines the protocol. The IPFIX protocol provides network administrators with access to IP Flow information. It was created based on the need for a common, universal standard of export for Internet Protocol flow information from routers, probes, and other devices that are used by mediation systems, accounting, or billing systems and network management systems to facilitate services such as measurement, accounting, and billing.

A Metering Process collects data packets at an Observation Point, optionally, filters them and aggregates information about these packets. Using the IPFIX protocol, an Exporter then sends this information to a Collector.
IPFIX is a push protocol, that is, each sender periodically send IPFIX messages to configured receivers without any interaction by the receiver. The actual makeup of data in IPFIX messages is to a great extent up to the sender. IPFIX introduces the makeup of these messages to the receiver with the help of special Templates. The sender is also free to use user-defined data types in its messages, so the protocol is freely extensible and can adapt to different scenarios. IPFIX prefers the Stream Control Transmission Protocol as its transport layer protocol, but also allows the use of the Transmission Control Protocol or User Datagram Protocol.

For more information about all the IPFIX entities, see [IP Flow Information Export (IPFIX) Entities](#).

**Related information:**


**IPFIX message format:**

An IPFIX Message consists of a Message Header, followed by one or more Sets. The Sets can be any of the possible three types - Data Set, Template Set, or Options Template Set.

**Note:** The Exporter codes all binary integers of the Message Header and the different Sets in network-byte order (also known as the big-endian byte ordering).

An IPFIX Message consisting of interleaved Template, Data, and Options Template Sets.

**IPFIX message header format**

Following are the message header field descriptions:

<table>
<thead>
<tr>
<th>Header field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Version of Flow Record format that is exported in this message. The value of this field is 0x000a for the current version, incrementing by one the version that is used in the NetFlow services export version 9</td>
</tr>
<tr>
<td>Length</td>
<td>Total length of the IPFIX Message, which is measured in octets, including Message Header and Sets.</td>
</tr>
<tr>
<td>Header field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Export Time</td>
<td>Time, in seconds, since 0000 Coordinated Universal Time Jan 1, 1970, at which the IPFIX Message Header leaves the Exporter.</td>
</tr>
<tr>
<td>Sequence Number</td>
<td>Incremental sequence counter-modulo $2^{32}$ of all IPFIX Data Records sent on this PR-SCTP stream from the current Observation Domain by the Exporting Process. Check the specific meaning of this field in the subsections of Section 10 when UDP or TCP is selected as the transport protocol. This value must be used by the Collecting Process to identify whether any IPFIX Data Records are missed. Template and Options Template Records do not increase the Sequence Number.</td>
</tr>
</tbody>
</table>
| Observation Domain ID | • A 32-bit identifier of the Observation Domain that is locally unique to the Exporting Process. The Exporting Process uses the Observation Domain ID to uniquely identify to the Collector.  
• Process the Observation Domain that metered the Flows. It is recommended that this identifier is unique per IPFIX Device. Collecting Processes must use the Transport Session.  
• Observation Domain ID field to separate different export streams that originate from the same Exporting Process. The Observation Domain ID must be 0 when no specific Observation Domain ID is relevant for the entire IPFIX Message. For example, when the Exporting Process Statistics are exported, or in a hierarchy of Collectors when aggregated Data Records are exported. |

**IPFIX Set format**

An IPFIX message consists of a message header followed by multiple Sets of different types. A Set is a generic term for collection of records that have a similar structure. There are three types of Sets - Data Set, Template Set, and Options Template Set. Each of these have a Set header and one or more records.

Every Set contains a common header. Following are the message header field descriptions:

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set ID</td>
<td>Set ID value identifies the Set. A value of 2 is reserved for the Template Set. A value of 3 is reserved for the Option Template Set. All other values 4-255 are reserved for future use. Values more than 255 are used for Data Sets. The Set ID values of 0 and 1 are not used for historical reasons</td>
</tr>
<tr>
<td>Header</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Length</td>
<td>Total length of the Set, in octets, including the Set Header, all records, and the optional padding. Because an individual Set MAY contain multiple records, the Length value must be used to determine the position of the next Set.</td>
</tr>
</tbody>
</table>

There are three types of sets:

- Data Set
- Template Set
- Options Template Set

**Related information:**


**NetFlow V9 Options Template:**

Some specific Options Templates and Options Template Records are necessary to provide extra information about the Flow Records and about the Metering Process.

An options template is a special type of template record that is used to communicate the format of data that is related to the NetFlow process. The options data record is a special type of data record (based on an options template) with a reserved template ID that provides information about the NetFlow process itself.

Rather than supplying information about IP flows, options are used to supply metadata about the NetFlow process itself. The format of the option template is as follows:
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlowSet ID = 1</td>
<td>The FlowSet ID is used to distinguish template records from data records. A template record always has a FlowSet ID of 1. A data record always has a nonzero FlowSet ID, which is greater than 255.</td>
</tr>
<tr>
<td>Length</td>
<td>This field gives the total length of this FlowSet. Because an individual template FlowSet might contain multiple template IDs, the length value must be used to determine the position of the next FlowSet record, which might be either a template or a data FlowSet. Length is expressed in TLV format, meaning that the value includes the bytes used for the FlowSet ID and the length bytes themselves, and the combined lengths of all template records included in this FlowSet.</td>
</tr>
<tr>
<td>Template ID</td>
<td>As a router generates different template FlowSets to match the type of NetFlow data it is exporting, each template is given a unique ID. This uniqueness is local to the router that generated the template ID. The Template ID is greater than 255. Template IDs inferior to 255 are reserved.</td>
</tr>
<tr>
<td>Option Scope Length</td>
<td>This field gives the length in bytes of any scope fields that are contained in this options template.</td>
</tr>
<tr>
<td>Options Length</td>
<td>This field gives the length (in bytes) of any Options field definitions that are contained in this options template.</td>
</tr>
<tr>
<td>Scope Field 1 Type</td>
<td>This field gives the relevant portion of the NetFlow process to which the options record refers. Currently, defined values follow: • 0x0001 System • 0x0002 Interface • 0x0003 Line Card • 0x0004 NetFlow Cache • 0x0005 Template For example, sampled NetFlow can be implemented on a per-interface basis, so if the options record was reporting on how sampling is configured, the scope for the report would be 0x0002 (interface).</td>
</tr>
<tr>
<td>Scope Field 1 Length</td>
<td>This field gives the length (in bytes) of the Scope field, as it would appear in an options record.</td>
</tr>
<tr>
<td>Option Field 1 Type</td>
<td>This numeric value represents the type of the field that appears in the options record.</td>
</tr>
<tr>
<td>Option Field 1 Length</td>
<td>This number is the length (in bytes) of the field, as it would appear in an options record.</td>
</tr>
</tbody>
</table>
Padding must be inserted to align the end of the FlowSet on a 32-bit boundary. Pay attention that the Length field includes those padding bits.

Options template example:

![Options Template Diagram]

Related information:


Default normalized flow record fields in Network Performance Insight

A default list of normalized Flow fields that are used with a common label across V9 and IPFIX flow formats and applicable for all vendors and protocols. Make sure to configure all the required fields in your Flow records.

Standard Flow fields

**Important:** These Flow fields that are stored by default in Network Performance Insight database. If you enable the option to store all Flow fields, all the additional IPFIX Flow entities are stored in the Hadoop-based file system and can be viewed with Apache Parquet Reader.

<table>
<thead>
<tr>
<th>Normalized Flow Fields</th>
<th>V9 Field name</th>
<th>IPFIX field name</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>packetSequence</td>
<td>=header(sequenceNumber)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exportTimestampMillis</td>
<td>=header(unixSeconds) * 1000 + (header(unixNSecs) / 1000000)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>baseTimestamp</td>
<td>= exportTimestampMillis - header(sysuptime)</td>
<td>= systemInitTimeMilliseconds</td>
<td></td>
</tr>
<tr>
<td>templateID</td>
<td>=header(sourceID) + templateID</td>
<td>=header(sourceID) + templateID</td>
<td></td>
</tr>
<tr>
<td>startTimestampMillis</td>
<td>=baseTimestamp + bytes(24-27)</td>
<td>= baseTimestamp + 21(LAST_SWITCHED)</td>
<td></td>
</tr>
<tr>
<td>Normalized Flow Fields</td>
<td>V9 Field name</td>
<td>IPFIX field name</td>
<td>Required</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------</td>
<td>-----------------</td>
<td>----------</td>
</tr>
<tr>
<td>endTimestampMillis</td>
<td>=baseTimestamp + bytes(28-31)</td>
<td>=baseTimestamp + 21-LAST_SWITCHED</td>
<td></td>
</tr>
<tr>
<td>inOctets</td>
<td>1 (IN_BYTES)</td>
<td>1 (octetDeltaCount)</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note: You must have either inOctets or outOctets in your Flow record.</td>
</tr>
<tr>
<td>inPackets</td>
<td>2 (IN_PKTS)</td>
<td>2 (packetDeltaCount)</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note: You must have either inPackets or outPackets in your Flow record.</td>
</tr>
<tr>
<td>protocolId</td>
<td>4 (PROTOCOL)</td>
<td>4 (protocolIdentifier)</td>
<td>Yes</td>
</tr>
<tr>
<td>srcTos</td>
<td>5 (SRC_TOS)</td>
<td>5 (ipClassOfService)</td>
<td></td>
</tr>
<tr>
<td>tcpBits</td>
<td>6 (TCP_FLAGS)</td>
<td>6 (tcpControlBits)</td>
<td></td>
</tr>
<tr>
<td>srcPort</td>
<td>7 (L4_SRC_PORT)</td>
<td>7 (sourceTransportPort)</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note: You must have srcPort or dstPort along with protocolId if applicationId field is not configured.</td>
</tr>
<tr>
<td>srcIp</td>
<td>8 (IPV4_SRC_ADDR)</td>
<td>8 (sourceIPv4Address)</td>
<td></td>
</tr>
<tr>
<td>srcMask</td>
<td>9 (SRC_MASK)</td>
<td>9 (sourceIPv4PrefixLength)</td>
<td></td>
</tr>
<tr>
<td>inIfId</td>
<td>10 (INPUT_SNMP)</td>
<td>10 (ingressInterface)</td>
<td>Yes</td>
</tr>
<tr>
<td>dstPort</td>
<td>11 (L4_DST_PORT)</td>
<td>11 (destinationTransportPort)</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note: You must have srcPort or dstPort along with protocolId if applicationId field is not configured.</td>
</tr>
<tr>
<td>dstIp</td>
<td>12 (IPV4_DST_ADDR)</td>
<td>12 (destinationIPv4Address)</td>
<td>Yes</td>
</tr>
<tr>
<td>dstMask</td>
<td>13 (DST_MASK)</td>
<td>13 (destinationIPv4PrefixLength)</td>
<td></td>
</tr>
<tr>
<td>outIfId</td>
<td>14 (OUTPUT_SNMP)</td>
<td>14 (egressInterface)</td>
<td>Yes</td>
</tr>
<tr>
<td>nextHopIp</td>
<td>15 (IPV4_NEXT_HOP)</td>
<td>15 (ipNextHopIPv4Address)</td>
<td></td>
</tr>
<tr>
<td>bgpSrcAsNum</td>
<td>16 (SRC_AS)</td>
<td>16 (bgpSourceAsNumber)</td>
<td></td>
</tr>
<tr>
<td>bgpDstAsNum</td>
<td>17 (DST_AS)</td>
<td>17 (bgpDestinationAsNumber)</td>
<td></td>
</tr>
<tr>
<td>bgpNextHopIp</td>
<td>18 (BGP_IPV4_NEXT_HOP)</td>
<td>18 (bgpNextHopIPv4Address)</td>
<td></td>
</tr>
<tr>
<td>endMs</td>
<td><em>baseTimestamp</em>21-LAST_SWITCHED</td>
<td>21 (flowEndSysUpTime)</td>
<td></td>
</tr>
<tr>
<td>startMs</td>
<td><em>baseTimestamp</em>22-FIRST_SWITCHED</td>
<td>22 (flowStartSysUpTime)</td>
<td></td>
</tr>
<tr>
<td>Normalized Flow Fields</td>
<td>V9 Field name</td>
<td>IPFIX field name</td>
<td>Required</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>------------------</td>
<td>----------</td>
</tr>
<tr>
<td>outOctets</td>
<td>23 (OUT_BYTES)</td>
<td>23 (postOctetDeltaCount)</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> You must have either inOctets or outOctets in your Flow record.</td>
</tr>
<tr>
<td>outPackets</td>
<td>24 (OUT_PKTS)</td>
<td>24 (postPacketDeltaCount)</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> You must have either inPackets or outPackets in your Flow record.</td>
</tr>
<tr>
<td>srcIp</td>
<td>27 (IPV6_SRC_ADDR)</td>
<td>27 (sourceIPv6Address)</td>
<td>Yes</td>
</tr>
<tr>
<td>dstIp</td>
<td>28 (IPV6_DST_ADDR)</td>
<td>28 (destinationIPv6Address)</td>
<td>Yes</td>
</tr>
<tr>
<td>srcMask</td>
<td>29 (IPV6_SRC_MASK)</td>
<td>29 (sourceIPv6PrefixLength)</td>
<td></td>
</tr>
<tr>
<td>dstMask</td>
<td>30 (IPV6_DST_MASK)</td>
<td>30 (destinationIPv6PrefixLength)</td>
<td></td>
</tr>
<tr>
<td>applicationId</td>
<td>95 (APPLICATION TAG)</td>
<td>95 (application tag)</td>
<td></td>
</tr>
<tr>
<td>policyQosClassification</td>
<td>41000 (policyQosClassificationHierarchy)</td>
<td>8232 (policyQosClassificationHierarchy)</td>
<td>These fields are required for QoS functionality and also for Top QoS Hierarchies with Queue ID aggregation to work correctly.</td>
</tr>
<tr>
<td>policyQosQueueId</td>
<td>42128 (Queue ID)</td>
<td>9360 (Queue ID)</td>
<td></td>
</tr>
<tr>
<td>flowDirection</td>
<td>61 (DIRECTION)</td>
<td>61 (flowDirection)</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If Application ID field is not available, then appName is resolved by using /etc/protocols and /etc/services as lookup tables or finding the service name based on a lookup from port and protocol fields from a NetFlow record. When no match is found, the appName field is populated with `<protocolId>:<lowerPortNumber>`.

Fields that are not part of the Flow data records but are enriched by the current Network Performance Insight solution are as follows:

- OUT_IF_ENTITY_ID
- IN_IF_ENTITY_ID
- OUT_IF_ENABLED
- IN_IF_ENABLED
- SRC_IP_GROUP
- DST_IP_GROUP

**Application monitoring**

*Table 4. NBAR fields (Derived from application name options template)*

<table>
<thead>
<tr>
<th>Normalized field name</th>
<th>V9 field name</th>
<th>IPFIX field name</th>
</tr>
</thead>
<tbody>
<tr>
<td>applicationDescription</td>
<td>94 (APPLICATION DESCRIPTION)</td>
<td>94 (applicationDescription)</td>
</tr>
</tbody>
</table>
Table 4. NBAR fields (Derived from application name options template) (continued)

<table>
<thead>
<tr>
<th>Normalized field name</th>
<th>V9 field name</th>
<th>IPFIX field name</th>
</tr>
</thead>
<tbody>
<tr>
<td>applicationID</td>
<td>95 (APPLICATION ID)</td>
<td>95 (applicationID)</td>
</tr>
<tr>
<td>applicationName</td>
<td>96 (APPLICATION NAME)</td>
<td>96 (applicationName)</td>
</tr>
</tbody>
</table>

Table 5. NBAR2 fields (Derived from application attributes options template)

<table>
<thead>
<tr>
<th>Normalized field name</th>
<th>V9 field name</th>
<th>IPFIX field name</th>
</tr>
</thead>
<tbody>
<tr>
<td>applicationCategoryName</td>
<td>45000 (application category name)</td>
<td>12232 (applicationCategoryName)</td>
</tr>
<tr>
<td>applicationSubCategoryName</td>
<td>45001 (application sub category name)</td>
<td>12233 (applicationSubCategoryName)</td>
</tr>
<tr>
<td>applicationGroupName</td>
<td>45002 (application group name)</td>
<td>12234 (applicationGroupName)</td>
</tr>
<tr>
<td>applicationBusinessRelevance</td>
<td>45012 (application business-relevance)</td>
<td>12244 (applicationBusinessRelevance)</td>
</tr>
<tr>
<td>p2p technology</td>
<td>288 (p2p technology)</td>
<td>288 (p2pTechnology)</td>
</tr>
<tr>
<td>tunnel technology</td>
<td>289 (tunnel technology)</td>
<td>289 (tunnelTechnology)</td>
</tr>
<tr>
<td>encrypted technology</td>
<td>290 (encrypted technology)</td>
<td>290 (encryptedTechnology)</td>
</tr>
</tbody>
</table>

Table 6. ART data fields

<table>
<thead>
<tr>
<th>Normalized field name</th>
<th>V9 field name</th>
<th>IPFIX field name</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxServerNwkTime</td>
<td>42088 (Server Network Time [max])</td>
<td>9320 (maxServerNwkTime)</td>
<td>You must configure your device for at least one of the fields.</td>
</tr>
<tr>
<td>maxClientNwkTime</td>
<td>42085 (Client Network Time [max])</td>
<td>9317 (maxClientNwkTime)</td>
<td></td>
</tr>
<tr>
<td>maxServerRespTime</td>
<td>42075 (Server Response Time [max])</td>
<td>9307 (maxServerRespTime)</td>
<td></td>
</tr>
<tr>
<td>maxTotalRespTime</td>
<td>42078 (Total Response Time [max])</td>
<td>9310 (maxTotalRespTime)</td>
<td></td>
</tr>
<tr>
<td>serverIPv4AddressOr</td>
<td>45005 (serverIPv4Address)</td>
<td>12237 (serverIPv4Address)</td>
<td>Yes</td>
</tr>
<tr>
<td>serverIPv6Address</td>
<td>45007 (serverIPv6Address)</td>
<td>12239 (serverIPv6Address)</td>
<td></td>
</tr>
<tr>
<td>ApplicationID</td>
<td>95 (APPLICATION ID)</td>
<td>95 (applicationId)</td>
<td>Yes</td>
</tr>
<tr>
<td>outIfID</td>
<td>14 (OUTPUT_SNMP)</td>
<td>14 (egressInterface)</td>
<td>Yes</td>
</tr>
<tr>
<td>ipDiffServCodePoint</td>
<td>195 (ipDiffServCodePoint)</td>
<td>195 (ipDiffServCodePoint)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

QoS monitoring

Table 7. QoS data fields

<table>
<thead>
<tr>
<th>Normalized field name</th>
<th>V9 field name</th>
<th>IPFIX field name</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>policyQoSQueueDrop</td>
<td>42129 (QoS Queue Drop)</td>
<td>9361 (policyQoSQueueDrop)</td>
<td>Yes</td>
</tr>
<tr>
<td>OutIfID</td>
<td>14 (OUTPUT_SNMP)</td>
<td>14 (egressInterface)</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 7. QoS data fields (continued)

<table>
<thead>
<tr>
<th>Normalized field name</th>
<th>V9 field name</th>
<th>IPFIX field name</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>policyQoSQueueID</td>
<td>42128 (Queue ID)</td>
<td>9360 (policyQosQueueId)</td>
<td>Yes</td>
</tr>
<tr>
<td>monitoringInterval</td>
<td>360 (monitoringInterval)</td>
<td>360 (monitoringInterval)</td>
<td>No</td>
</tr>
<tr>
<td>EndMilliSeconds</td>
<td>EndMilliSeconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Note: This field is not configured in your device, use the timestamp from the received packet.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8. classmap options template table fields

<table>
<thead>
<tr>
<th>Normalized field name</th>
<th>V9 field name</th>
<th>IPFIX field name</th>
</tr>
</thead>
<tbody>
<tr>
<td>classId</td>
<td>41001 (c3pl class cce-id)</td>
<td>8233 (classId)</td>
</tr>
<tr>
<td>className</td>
<td>41002 (c3pl class name)</td>
<td>8234 (className)</td>
</tr>
<tr>
<td>classType</td>
<td>41003 (c3pl class type)</td>
<td>8235 (classType)</td>
</tr>
</tbody>
</table>

Table 9. policymap options template table fields

<table>
<thead>
<tr>
<th>Normalized field name</th>
<th>V9 field name</th>
<th>IPFIX field name</th>
</tr>
</thead>
<tbody>
<tr>
<td>policyID</td>
<td>41004 (c3pl policy cce-id)</td>
<td>8236 (policyID)</td>
</tr>
<tr>
<td>policyName</td>
<td>41005 (c3pl policy name)</td>
<td>8237 (policyName)</td>
</tr>
<tr>
<td>policyType</td>
<td>41006 (c3pl policy type )</td>
<td>8238 (policyType)</td>
</tr>
</tbody>
</table>

For more information about Flow fields, see the following sections:
- Enabling additional Flow raw fields collection in Installing and Configuring IBM Network Performance Insight
- Determine the absolute time for the Flows in a NetFlow IPFIX data packet in Troubleshooting IBM Network Performance Insight
- Configuring Flexible NetFlow and AVC in IBM Network Performance Insight: References

Flow Analytics

The Flow Analytics Service computes Top-N aggregations for 1 minute, 30 minutes, and 1-day intervals and also evaluates thresholds on interface usage. It provides Traffic Ingress and Egress details of every interface level.

The Flow Analytics Service is also responsible for aggregation processes. It refines the raw data, filters the results, and aggregates the KPI values. The values are aggregated by SUM, and the results are then stored in Network Performance Insight database.

Flow Analytics Service aggregates the data every 1 minute, 30 minutes, and 1-day intervals. For 1-minute aggregation, it is distributed based on record segmentation and the aggregation is based on SUM of RAW data. For 30 minutes and 1-day aggregation, it is distributed based on the aggregation type. The aggregation for 30 minutes is based on SUM of 1-minute data and for daily aggregation is based on SUM of 30-minutes data.
The Flow Analytics Service also triggers the IP to Domain Name, and Domain Name to IP resolution to DNS service.

Flow Analytics Service provides these basic functions:
- Categorizes, aggregates, and ranks the data that is collected by the Collector Service.
- Detects and alerts real-time threshold violations. A threshold is a value that is compared against metrics to determine whether the metrics exceed or drop below a specific constraint. Currently, only burst threshold is supported. Burst threshold ignores the natural network bursts by evaluating how long in a row the violations occurred per resource. Burst thresholds can be set and reset multiple times.

**Related concepts:**
“DNS” on page 9

The DNS Service resolves DNS names for reporting and distributing the interface metadata.

**Built-in aggregation definitions**
This section details the built-in aggregation types and their grouping keys. Based on these aggregations the data for the Top Talker views from the Traffic Details dashboard is populated. These aggregations are user configurable.

<table>
<thead>
<tr>
<th>Aggregation name</th>
<th>Grouping keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Applications</td>
<td>IF_ID + APP_NAME</td>
</tr>
<tr>
<td>Top Applications with Source ToS</td>
<td>IF_ID + APP_NAME+SRC_TOS</td>
</tr>
<tr>
<td>Top IP Group Conversations with Application</td>
<td>IF_ID +SRC_IP_GROUP+DST_IP_GROUP+APP_NAME</td>
</tr>
<tr>
<td>Top IP Group Conversations with Protocol</td>
<td>IF_ID +SRC_IP_GROUP+DST_IP_GROUP+PROTOCOL_ID</td>
</tr>
<tr>
<td>Top IP Group Conversations with Source ToS</td>
<td>IF_ID +SRC_IP_GROUP+DST_IP_GROUP+SRC_TOS</td>
</tr>
<tr>
<td>Top Conversations with Application</td>
<td>IF_ID+SRC_IP+DST_IP+APP_NAME</td>
</tr>
<tr>
<td>Top Autonomous System Conversations</td>
<td>IF_ID+BGP_SRC_AS_NUM+BGP_DST_AS_NUM</td>
</tr>
<tr>
<td>Top Conversations</td>
<td>IF_ID+SRC_IP+DST_IP</td>
</tr>
<tr>
<td>Top Conversations with ToS</td>
<td>IF_ID+SRC_IP+DST_IP+SRC_TOS</td>
</tr>
<tr>
<td>Top Destination Autonomous System</td>
<td>IF_ID+BGP_DST_AS_NUM</td>
</tr>
<tr>
<td>Top Destination IP Groups with Application</td>
<td>IF_ID+DST_IP_GROUP+APP_NAME</td>
</tr>
<tr>
<td>Top Destination IP Groups with Protocol</td>
<td>IF_ID+DST_IP_GROUP+PROTOCOL_ID</td>
</tr>
<tr>
<td>Top Destination IP Groups with Source ToS</td>
<td>IF_ID+DST_IP_GROUP+SRC_TOS</td>
</tr>
<tr>
<td>Top Destinations with Application</td>
<td>IF_ID+DST_IP+APP_NAME</td>
</tr>
<tr>
<td>Top Destinations</td>
<td>IF_ID+DST_IP</td>
</tr>
<tr>
<td>Top QoS Hierarchies with Queue ID</td>
<td>IF_ID+POLICY_QOS_CLASSIFICATION_HIERARCHY+POLICY_QOS_QUEUE_ID</td>
</tr>
<tr>
<td>Top Protocols with Application</td>
<td>IF_ID+PROTOCOL_ID+APP_NAME</td>
</tr>
<tr>
<td>Top Protocols with Conversation</td>
<td>IF_ID+PROTOCOL_ID+SRC_IP+DST_IP</td>
</tr>
<tr>
<td>Top Protocols with Destination IP</td>
<td>IF_ID+PROTOCOL_ID+DST_IP</td>
</tr>
<tr>
<td>Top Protocols</td>
<td>IF_ID+PROTOCOL_ID</td>
</tr>
<tr>
<td>Top Protocols with Source IP</td>
<td>IF_ID+PROTOCOL_ID+SRC_IP</td>
</tr>
<tr>
<td>Top Source Autonomous System</td>
<td>IF_ID+BGP_SRC_AS_NUM</td>
</tr>
<tr>
<td>Aggregation name</td>
<td>Grouping keys</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Top Source IP Groups with Application</td>
<td>IF_ID+SRC_IP_GROUP+APP_NAME</td>
</tr>
<tr>
<td>Top Source IP Groups with Protocol</td>
<td>IF_ID+SRC_IP_GROUP+PROTOCOL_ID</td>
</tr>
<tr>
<td>Top Source IP Groups with Source ToS</td>
<td>IF_ID+SRC_IP_GROUP+SRC_TOS</td>
</tr>
<tr>
<td>Top Sources</td>
<td>IF_ID+SRC_IP</td>
</tr>
<tr>
<td>Top Source with Application</td>
<td>IF_ID+SRC_IP+APP_NAME</td>
</tr>
<tr>
<td>Top Source ToS</td>
<td>IF_ID+SRC_TOS</td>
</tr>
</tbody>
</table>

**Threshold**

A threshold is a value that is compared against the predefined threshold configurations. It is evaluated to see whether it violates a specific restriction. The primary objective of thresholding is to determine any violations and to generate alerts. When the value falls outside the acceptable threshold range, the system generates and stores the event condition and forwards it to the Event Management System.

**Static thresholds**

Static (Burst) thresholding is user-defined static values at specific intervals, which analyze data and generate events when a violation occurs.

If your IBM Netcool Operations Insight solution is integrated with Network Performance Insight, then you can define static thresholds for anomaly detection.

You can define a static threshold for a KPI within the poll definition that polls for that KPI. If these static thresholds are violated for any performance measure on a device or interface, IBM Tivoli Netcool/OMNibus events are generated at an appropriate severity level.

Thresholds define the status of an attribute based on specific conditions. You can enable threshold evaluation on a selected resource or interface. A threshold is violated when the result of the collected metric value is evaluated as exceeding (upper) or dropping (lower) to a specified configured threshold level. The actual evaluation and disposition depends on the threshold type, Upper, Lower, or Band.

To define an anomaly threshold, see [Defining anomaly thresholds](#).

**Threshold violation:**

A KPI is considered to be in a violation state when it breaches the limits of traffic volume data for a specific number of times according to the configuration settings.

**Threshold type**

The severity of a violation is decided based on the violation against either the upper or lower limit that is configured for a threshold.

- If the threshold is configured to check for Over, then severity is determined as:
  - If KPI value is greater than or equal to the Upper Limit, then the severity is Critical.
* If the KPI value is greater than or equal to the lower limit, but not exceeding the upper limit, then the severity is Major.

* If the threshold is configured to check for Under, then severity is determined as:
  * If the KPI value is lesser than or equal to the upper limit, but not dropping below the lower limit, then the severity is Major.
  * If KPI value is lesser than or equal to the lower limit, then the severity is Critical.

* If the threshold is configured to check for Band, then severity is determined as:
  * If the KPI value is greater than or equal to the upper limit, then the severity is Critical.
  * If KPI value is lesser than or equal to the lower limit, then the severity is Critical.

For example, if the limit type is Over, and the traffic volume data is greater than the upper limit, the severity of the threshold is set to Critical. If the traffic volume data is greater than the lower limit, the threshold severity is set to Major.

The following table summarizes how a threshold severity is set to either Major or Critical for different limit types:

**Table 11. Threshold severity for limit types**

<table>
<thead>
<tr>
<th>Limit Type</th>
<th>Scenario</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Over</td>
<td>Traffic volume data greater than the upper limit</td>
<td>Critical</td>
</tr>
<tr>
<td></td>
<td>Traffic volume data greater than the lower limit</td>
<td>Major</td>
</tr>
<tr>
<td>Under</td>
<td>Traffic volume data lesser than upper limit</td>
<td>Major</td>
</tr>
<tr>
<td></td>
<td>Traffic volume data lesser than the lower limit</td>
<td>Critical</td>
</tr>
<tr>
<td>Band</td>
<td>Traffic volume data greater than the upper limit</td>
<td>Critical</td>
</tr>
<tr>
<td></td>
<td>Traffic volume data lesser than the lower limit</td>
<td>Critical</td>
</tr>
</tbody>
</table>

**Note:** For limit type Band, the severity is always set to Critical for any threshold violation scenarios.

**Exiting a threshold violation**

Exiting from a violation state is the action of setting the severity of an event to CLEAR. The severity is cleared after a KPI goes into a non-violation state. The threshold at which an event is raised then is cleared.

Non-Violation -> Major Violation -> CLEAR -> Non-Violation

Or

Non-Violation -> Critical Violation -> CLEAR -> Non-Violation

Network Performance Insight supports system generated auto clear. The following table shows, when the system can set the severity to CLEAR.
Table 12. Scenarios for CLEAR severity.

Table that shows scenarios when CLEAR severity occurs

<table>
<thead>
<tr>
<th>Limit Type</th>
<th>Current Severity</th>
<th>Scenario</th>
<th>New Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Over</td>
<td>KPI is either in Major or Critical level</td>
<td>Traffic volume data lesser than the lower limit</td>
<td>CLEAR</td>
</tr>
<tr>
<td>Under</td>
<td>KPI is either in Major or Critical level</td>
<td>Traffic volume data greater than upper limit</td>
<td>CLEAR</td>
</tr>
<tr>
<td>Band</td>
<td>KPI is in Critical level</td>
<td>Traffic volume data falls within the range of upper limit and lower limit</td>
<td>CLEAR</td>
</tr>
</tbody>
</table>

If an existing threshold configuration is modified, by changing any of the configuration values such as limit type, upper or lower limit, number of events or disable the threshold, the system sets the severity to CLEAR.

For more information about the configuration steps, see Configuring Thresholds from Console Integrations in Using Network Performance Insight.

Remote Flow Collector

Install Remote Flow Collector if you want the collector to be co-located with the Flow exporters from which it is collecting data. The Flow records that are collected by the Remote Collectors are sent to the Ambari agent hosts where the Storage Service is located.

It performs the same functions as a Flow Collector Service does. The ratio between remote and local collectors must be 1:1.

For more information about how to install and control the Remote Flow Collector Service, see Installing and setting up Remote Flow Collector section in Installing and Configuring IBM Network Performance Insight.

Performance Metric OOTB Device Support

Performance Metric OOTB Device Support solution enhances the capability of Tivoli Network Manager to discover and poll certain built-in vendor-specific device performance metrics. The configured Storm Spout then pushes the polled metrics from NCPOLLDATA database to Network Performance Insight Kafka message bus. From the Kafka topics, the data is stored in Network Performance Insight database and used for visualization.

The Performance Metric OOTB Device Support provides the performance metrics with predefined formulas for the following vendor devices and their supported MIB files:

- Cisco
- Huawei
- Juniper
With the help of Performance Metric OOTB Device Support, the device monitoring
capability of Network Performance Insight is enhanced. It provides detailed
device or infrastructure performance metrics, topology visualization, and root cause
analysis.

**Performance Metric OOTB Device Support mechanism**

Performance Metric OOTB Device Support solution inserts, updates, and maps the
polled device performance metrics that are discovered with the help of
vendor-specific MIB files. It is able to provide information for a metric at an entity
level. The information is then reported on the Network Performance Insight
Dashboards.

Performance Metric OOTB Device Support solution works as follows:

- Maps the new metrics with the vendor-specific MIB OIDs. A set of additional
  MIB files that are not available in Tivoli Network Manager system are supplied
  in Performance Metric OOTB Device Support package.
- Discovers and stores the entity level information of a KPI. It works as follows:
  Even though the entities are discovered by Tivoli Network Manager discovery
  agent and the individual Entity IDs are visible in NCIM.ENTITYDATA table, the
  Entity IDs are not populated in NCPOLLDATA database. Performance Metric OOTB
  Device Support maps the Entity IDs and extracts the individual entity labels and
  populates the Entity Names for the new metrics in NCIM.ENTITYDATA table. From
  the NCIM.ENTITYDATA, it extracts the entity IDs for those metrics and populates
  the individual Entity IDs for those metrics in NCPOLLDATA.MONITOREDINSTANCE
  table.
  Also, if the discovery record is not found in the NCPOLLDATA.MONITOREDENTITY
  table, Performance Metric OOTB Device Support solution inserts the selected
  records to NCPOLLDATA.MONITOREDENTITY. It uses the IP address and the Entity ID
  of the particular metric from NCIM.ENTITYDATA table.

For more information about Performance Metric OOTB Device Support solution,
see the following sections:

- Installing the Performance Metric OOTB Device Support component from Installing
  and Configuring IBM Network Performance Insight.
- Supported Performance Metric OOTB Device Support formulas from IBM Network
  Performance Insight: References
- Network Performance Overview dashboard and On-demand dashboard sections from
  Using IBM Network Performance Insight

**Related information:**

- [Enabling the integration with Network Performance Insight](#)
- [Tivoli Network Manager database reference](#)
Network Performance Insight Dashboards

Network Performance Insight Dashboards are powerful and highly customizable dashboards that provide a shape and structure to your network performance data. These dashboards are suitable for Capacity Planners, Operations Engineers, and Operations Managers to analyze and report the performance and health of network resources and this data can then be used to correlate and pinpoint the network issues for troubleshooting.

Features of Network Performance Insight Dashboards

When you install Network Performance Insight V1.2.2, Network Performance Insight Dashboards are pre-loaded. You can display your raw and aggregated data that is collected, analyzed, and stored by Network Performance Insight in cross-sectional and timeseries representations. Network Performance Insight Dashboards are based on the following features:

• Simple, modern, and flexible platform with an interactive widget library that can integrate with multiple data sources.
• Consist of summary, correlation, and detailed level dashboards that are set with master, listener, and drill-down relations.
• Provide rich interactivity as follows:
  – Custom filters
  – Zoom in and out capabilities
  – Interchangeable widget types
  – Changed correlations and drill-down to details with focus change
• Co-exist with other Networks for Operation Insight dashboards; Network Health Dashboard and Device Dashboard.
• Provide role-based access to different dashboards.
• Export and share the report information with other stakeholders in PDF, CSV, or XLS formats.

Types of Network Performance Insight Dashboards

Network Performance Overview dashboard

Provides summary-level information about network performance and health and helps you understand the context to navigate to more specific areas or historical performance of particular device. Comprehensive network performance information is provided under the following areas:

• Congestion
• Quality of Service
• Traffic
• Device Load

NetFlow dashboards

These dashboards render the metrics that represent the IP network traffic information that is collected from the Flow records that enter or exit an interface. Specific Flexible NetFlow metrics and Application Visibility and Control metrics that are collected, aggregated, and stored in Network Performance Insight are displayed as top talker resources. These detail-level dashboards can help troubleshoot NetFlow issues.

On Demand Filtering dashboards

The IP SLA on-demand dashboards can help investigate the performance
of a specific interface over a period based on a set of metrics. These dashboards provide correlations from the other dashboards to narrow down the issue to a device and entity from the historical trends on specific IP SLA metrics. The On Demand Filtering dashboards provide the following types of data:

- IP SLA
- Device Health

For more information about these dashboards, see Network Performance Insight Dashboards section in Using IBM Network Performance Insight.
Chapter 3. Deployment considerations

All Network Performance Insight components must be installed on Red Hat Linux, Version 7.2.

Note: For fresh installation of Network Performance Insight V1.2.2 on RHEL 7.4 platform, apply [1.2.2.0-TIV-NPI-IF0001] and then install 1.2.2.

Co-location rules

While it is possible to deploy all the Network Performance Insight and its associated components on a single instance for evaluation purpose. Typically, you must have at least three hosts; one master Ambari server, and two Ambari agent slaves for Network Performance Insight cluster. Consider the following scenarios:

- You must plan for servers to install Jazz for Service Management and other Netcool Operations Insight components.
- You can integrate Network Performance Insight that is always installed with root user with and other supported Netcool Operations Insight components that are installed with non-root user.

Related information:

Performing a fresh installation

Suggested node and services layout

Use the following guidelines for setting up your IBM Open Platform with Apache Spark and Apache Hadoop and Network Performance Insight V1.2.2 services in your cluster.

Ambari agent deployment deploys Network Performance Insight service layer and application binary to the cluster hosts, and installs each component to the default location (/opt/IBM/npi or /opt/IBM/basecamp) and the IBM Open Platform with Apache Spark and Apache Hadoop components to /usr/iop/current directory.

For more information, see Installation directory structure section in Installing and Configuring IBM Network Performance Insight.

Multi-node cluster deployment

It is suggested that you have at least one Ambari server node and the rest of them as Ambari agents. In the diagram, HOST A is the Ambari server and HOST B, C, and D are the Ambari agents.
Note:

- Make sure that you install Manager Service and Kafka Broker in all Ambari agent nodes. Kafka Schema Registry must be installed along with Kafka Broker.
- Because Zookeeper requires a majority, it is best to use an odd number of machines. For example, with four machines ZooKeeper can handle the failure of a single machine; if two machines fail, the remaining two machines do not constitute a majority. However, with five machines ZooKeeper can handle the failure of two machines.
- One Flow Collector can support only one Remote Flow Collector.

See Netcool Operations Insight documentation for installation scenario for other integrated components.

Related information:

- [Deployment considerations](#)
- [Suggested services layout for IBM Open Platform with Apache Spark and Apache Hadoop and BigInsights value-added services](#)
Cluster behavior

Provides the relevance between Network Performance Insight and its related services with the node behavior in a cluster.

Network Performance Insight supports the following types of node behavior.

**Cluster singleton**
A clustered singleton service (also known as an HA singleton) is a service that is deployed on multiple nodes in a cluster, but is providing its service on only one of the nodes. The node that is running the singleton service is typically called the oldest node.

**Load balancing**
Load balancing improves the distribution of workloads across multiple nodes where each of the node serves different set of clients that are mutually exclusive.

**Managed load balancing**
The difference between Load Balancing with Managed load balancing here is that, node acts as manager node to monitor the load balancing activities. The manager node monitors and distributes the workload among the active nodes.

**Data replication**
A replication strategy determines the nodes where data replicas are placed. The replicas on multiple nodes are stored to ensure reliability and fault tolerance.

**Monitoring in each node**
A service that is installed on each node in a cluster, where it monitors and provides information on the installed nodes.

**Single instance**
A service that is installed on a single node in a cluster, which provides its service across all nodes.

The following table lists the service components and their node behavior. Use the following information as guidance to set up your environment.
<table>
<thead>
<tr>
<th>Services</th>
<th>Service Type</th>
<th>Service Components</th>
<th>Cluster node behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Performance Insight</td>
<td>Master</td>
<td>Manager</td>
<td>Monitoring on each node</td>
</tr>
<tr>
<td>Slave</td>
<td>DNS</td>
<td></td>
<td>Cluster singleton</td>
</tr>
<tr>
<td>Slave</td>
<td>Entity Analytics</td>
<td></td>
<td>Cluster singleton</td>
</tr>
<tr>
<td>Slave</td>
<td>Event</td>
<td></td>
<td>Cluster singleton</td>
</tr>
<tr>
<td>Slave</td>
<td>Flow Analytics</td>
<td></td>
<td>Managed load balancing</td>
</tr>
<tr>
<td>Slave</td>
<td>Flow Collector</td>
<td></td>
<td>Load balancing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see Configuring the number of interfaces in Installing and Configuring IBM Network Performance Insight.</td>
</tr>
<tr>
<td>Slave</td>
<td>Tivoli Network Manager Collector</td>
<td></td>
<td>Cluster singleton</td>
</tr>
<tr>
<td>Slave</td>
<td>Storage</td>
<td></td>
<td>Cluster singleton</td>
</tr>
<tr>
<td>Slave</td>
<td>Threshold</td>
<td></td>
<td>Cluster singleton</td>
</tr>
<tr>
<td>Slave</td>
<td>UI</td>
<td></td>
<td>Load balancing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see UI Service in IBM Network Performance Insight: Product Overview.</td>
</tr>
<tr>
<td>Slave</td>
<td>SNMP Collector</td>
<td></td>
<td>Cluster singleton</td>
</tr>
<tr>
<td>Slave</td>
<td>Formula Service</td>
<td></td>
<td>Cluster singleton</td>
</tr>
<tr>
<td>HDFS</td>
<td>Master</td>
<td>NameNode</td>
<td>Single instance</td>
</tr>
<tr>
<td>Slave</td>
<td>SNameNode</td>
<td></td>
<td>Data replication</td>
</tr>
<tr>
<td>Slave</td>
<td>DataNode</td>
<td></td>
<td>Data replication</td>
</tr>
<tr>
<td>YARN</td>
<td>Master</td>
<td>Timeline Server</td>
<td>Single instance</td>
</tr>
<tr>
<td>Slave</td>
<td>Resource Manager</td>
<td></td>
<td>Single instance</td>
</tr>
<tr>
<td>Slave</td>
<td>Node Manager</td>
<td></td>
<td>Managed load balancing</td>
</tr>
<tr>
<td>ZooKeeper</td>
<td>Master</td>
<td>ZooKeeper</td>
<td>Data replication</td>
</tr>
<tr>
<td>Ambari Metrics</td>
<td>Master</td>
<td>Collector</td>
<td>Single instance</td>
</tr>
<tr>
<td>Slave</td>
<td>Monitor</td>
<td></td>
<td>Monitoring on each node</td>
</tr>
<tr>
<td>Kafka</td>
<td>Master</td>
<td>Kafka Broker</td>
<td>Data replication</td>
</tr>
<tr>
<td></td>
<td>Master</td>
<td>Kafka Connect</td>
<td>Single instance</td>
</tr>
<tr>
<td></td>
<td>Master</td>
<td>Kafka Schema Registry</td>
<td>Load balancing</td>
</tr>
</tbody>
</table>
Table 13. Cluster node behavior (continued)

<table>
<thead>
<tr>
<th>Services</th>
<th>Service Type</th>
<th>Service Components</th>
<th>Cluster node behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>MapReduce2</td>
<td>Master</td>
<td>History Server</td>
<td>Single instance</td>
</tr>
</tbody>
</table>
Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those
websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

The client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

**Trademarks**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

**Terms and conditions for product documentation**

Permissions for the use of these publications are granted subject to the following terms and conditions.

**Applicability**

These terms and conditions are in addition to any terms of use for the IBM website.

**Personal use**

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

**Commercial use**

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

**Rights**

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

*IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.*