User Guide

Arista Analytics

Version 8.8

Arista.com

Headquarters	Support	Sales
5453 Great America Parkway Santa Clara, CA 95054 USA +1-408-547-5500	+1-408-547-5502 +1-866-476-0000	+1-408-547-5501 +1-866-497-0000
www.arista.com/en/	support@arista.com	sales@arista.com

Contents

Chapter 1

Introduction

Arista Analytics Node provides scale-out analytics with configurable, historical time-series-based dashboards for flow visibility, health, performance, and capacity planning. It acts as a collector for NetFlow and sFlow packets to provide real-time visibility, including tunneled or encapsulated traffic, enabling the detection of security attacks like DoS/DDoS and SYN attacks. The highly intuitive and customizable GUI dashboards support a search to drill down and focus on possible issues quickly. It provides a variety of reporting and alerting functions and allows the user to easily share custom dashboard views with other team members for collaborative analysis, troubleshooting, and remediation.

1.1 Analytics and Dashboards

Arista Analytics provides the accessibility to analyze, search, predict, and reveal patterns and relationships among data.

The following options to access Arista Analytics features:

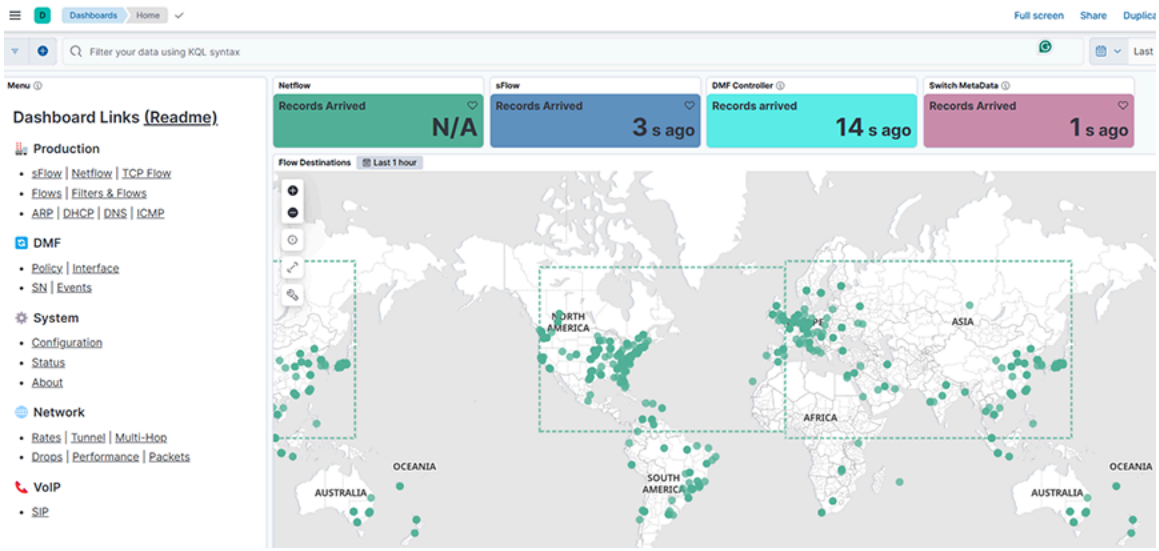
- **Dashboard**: Monitors traffic flow, network health, performance, and capacity planning. There is an option to customize dashboards.
- **Discover**: Use predefined indices to filter and display specific events.
- **Machine Learning**: Detect, model, and predict unusual activity or events on the network.

The Arista Analytics displays in the following tabs:

- **Production Network**: This analyzes the main operational network.
- **DMF**: This is for observing the dedicated monitoring network.
- **System**: This configures the analytics system itself.
- **Network**: This manages the analytics system.

- **VoIP**: This configures the analytics system itself.

Figure 1-1: Production Network > Dashboard



Each tab utilizes panels that display data through:

- Visualizations (pie charts, line graphs, etc.) based on queries.
- An event list at the bottom shows matching events.
- Pop-up windows for detailed information on panel mouseovers.

It provides a clear structure for monitoring and managing network performance and system settings.

The Kibana documentation documents the Analytics GUI, and most of its features and operations based on ElasticSearch are available at the following URL:

https://www.elastic.co/guide/en/kibana/8.15/index.html

1.2 Common Features and Functions

The Arista Analytics displays in the following visualizations:

- **Area**: Emphasize the data between an axis and a line
- **Data Table**: Displays data in rows and columns.
- **Heat map**
- **Horizontal bar**
- **Line**
- **Metric**
- **Recorder Node**
- **Tag Cloud**
- **Timelion**

- **Vertical bar**

1.2.1 Filtering Information on a Dashboard

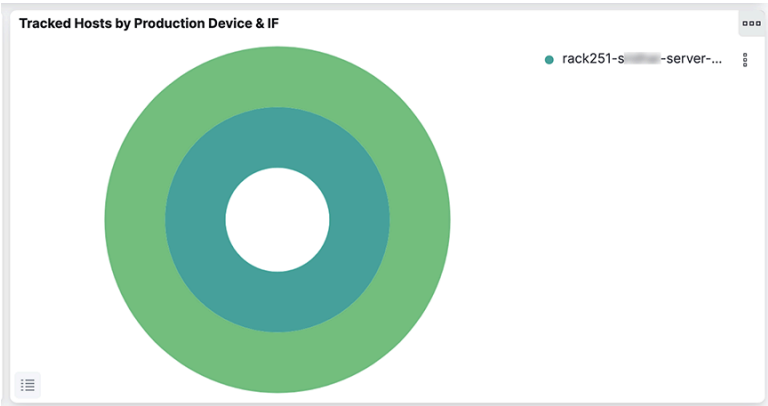
The interactive filtering capabilities of the Arista Analytics dashboard:

- **Area-Based Filtering**: Selecting an area on the dashboard restricts the displayed events to those similar to the selected area. It allows for context-sensitive filtering.
- **Pie Chart Slice Filtering**: Clicking a slice of a pie chart filters the dashboard to show only events related to the specific activity represented by that slice. It is a direct way to isolate and examine particular activities.
- **Color Customization**: Users can change the color associated with protocols or other objects by clicking their labels in the list beside the chart. It enhances visual clarity and allows for personalized data representation.

1.2.2 Pie

Pie charts that display information by the production switch have an inner and outer ring, as shown in the following example.

Figure 1-2: Two-ring Pie Chart



For example, in the **Tracked Hosts by Production Device & IF** pie chart,

the detailed behavior of the pie charts in the Arista Analytics Fabric view, specifically those displaying information related to production switches, is shown. The key feature is the dual-ring structure:

- **Inner Ring:** Represents a broader category, "Production Device" (switches).
- **Outer Ring:** Provides a more granular breakdown, such as "Interface" (IF) details selected inner ring segment.

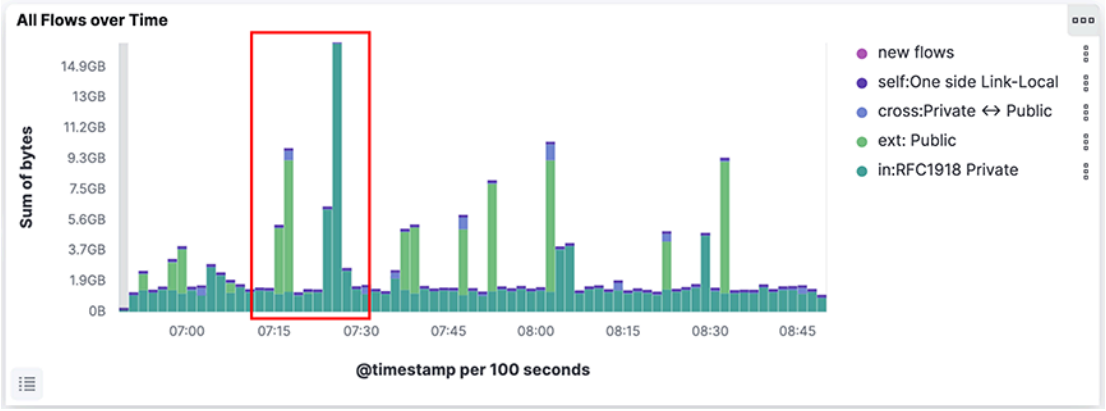
The interactive functionality is crucial, as clicking a segment in the inner ring filters the outer ring to display only the data relevant to that selected inner ring segment.

It allows for a hierarchical view of the data, enabling users to quickly drill down from a general overview (switch level) to specific details (interface level) within that overview. It is a good design for exploring relationships within the network data.

1.2.3 Selecting the Time Range

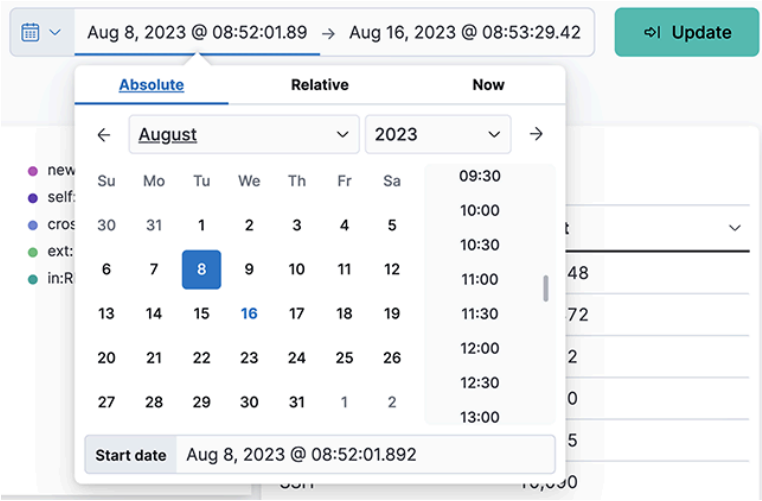
To restrict the current content to events occurring in a specific period, click and drag it to surround the area on a time visualization, such as the Flows Over Time.

Figure 1-3: Selecting the Time Range



To select the time range or to change the default refresh rate, click the **Time Range** control in the upper right corner. The system displays the following dashboard.

Figure 1-4: Time Range Control

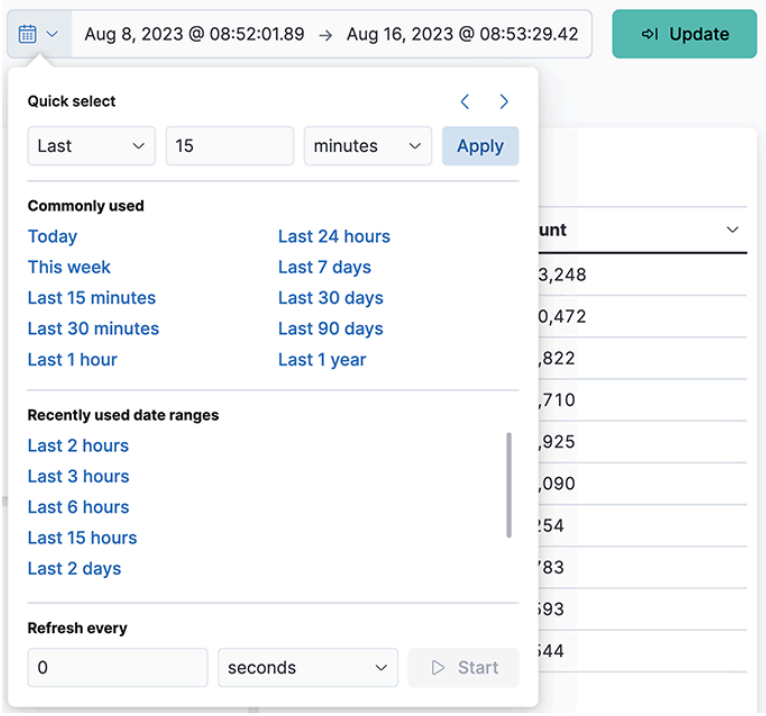


This dialog provides the following options for setting the time range:

- **Quick**: Simple settings, such as Today, Last 1 hour, etc.
- **Relative**: Time offsets from a specific time, including the current time.
- **Absolute**: Set a range based on date and time.
- **Recent**: Provides a list of recently used ranges that you can reuse.

Select the range from the options provided, and the panels and displays update to reflect the new date and time range. To change the auto-refresh rate, click the **Auto-refresh** control. The system displays the following dashboard.

Figure 1-5: Change Auto Refresh Rate

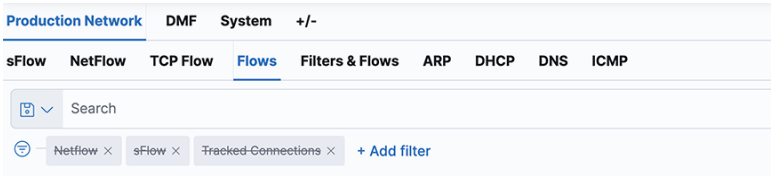


Select the refresh interval from the options provided. Click **Start** to turn off the auto-refresh function.

1.2.4 Using the Search Field

The search field at the top of the dashboard filters the current displays by any text or numbers typed into the field.

Figure 1-6: Search Field



The green bars under the **Search** field show the currently applied filters. When the pointer is over a green bar, it displays icons that control the filter.

- **Enable/Disable filter**
- **Pin/Unpin filter**
- **Exclude/Include matches**
- **Remove filter**
- **Edit filter**

The **Action** option in the upper right corner applies these actions to all the currently applied filters.

Click a segment on a pie chart for the appropriate filter; it automatically inserts into the **Search** field. To undo the filter, click the **Remove** filter icon.

To filter the information in the displays, enter the characters to filter the display in the search field. For example, for entering the first part of an IP address, it updates the displays to show only those IP addresses that match the characters entered. The following are some of the most helpful search filters:

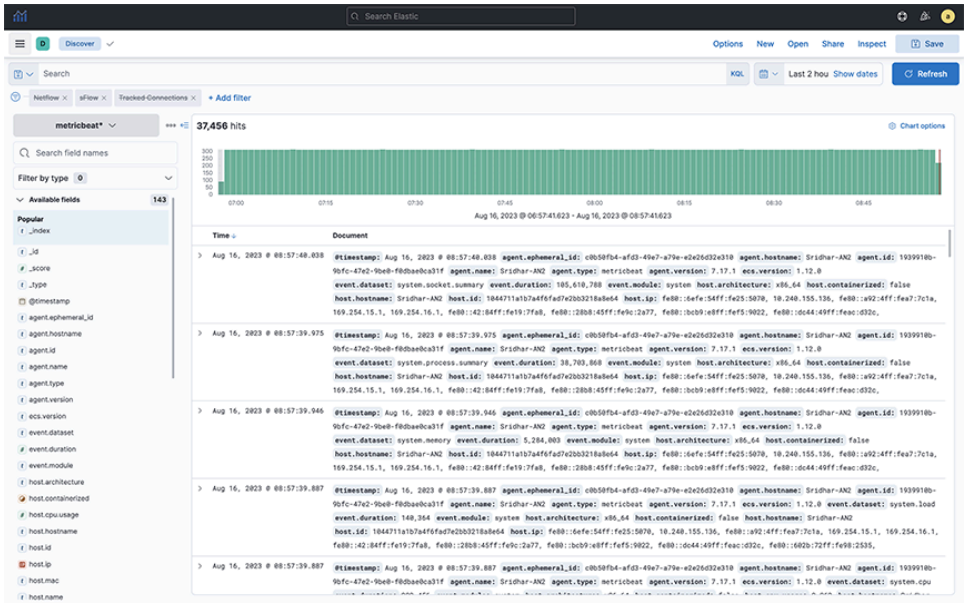
- IP address
- Host name (requires DNS services)
- Protocol, for example, HTTP, HTTPS, ICMP, and so forth
- DMF interface name

To define complex queries using field names, which can be seen by scrolling and clicking on an event row. For example, on the sFlow®* dashboard, the query `proto : TCP AND tags : ext` displays all externally bound TCP traffic. OR NOT ( ) are also permitted in the expression. For more details about the supported search syntax, refer to the following URL:https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-query-string-query.html#query-string-syntax.

1.2.5 Using Discover Mode

Select the **Discover** option in the left panel of the **Analytics** window; the system will display the following page.

Figure 1-7: Discover Mode



Use **Discover** mode to see the indices in the ElasticSearch database and identify the available data.

* sFlow® is a registered trademark of the Inmon Corp.

1.2.6 Search Performance Limitations

Refrain from executing general queries for a wide time range. For example, suppose you want to query for 7 or 30 days. In that case, do a specific query flow, filter interface, specific source or destination IP address, and specific source or port number as it eases the query load.

To query NetFlow or sFlow® for more extended periods, use the **FLOW** Dashboard to determine the trend and then do a specific query, such as querying a specific flow or time, on the Netflow or sFlow® dashboard.

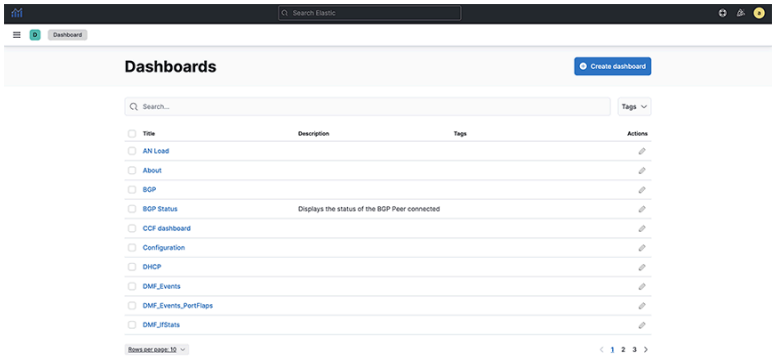
Periodically monitor the **AN Load** Dashboard for index and query load and scale up the Analytics Node if the Load is Yellow/Red.

These recommendations aim to optimize query performance and prevent system overload, especially when dealing with large datasets and extended timeframes.

1.2.7 Managing Dashboards

Select the **Dashboards** option from the left panel on the **Analytics** window to manage Dashboards. The system displays the following page.

Figure 1-8: Dashboard Mode



Refer to the Kibana documentation for details about creating and managing dashboards.https://www.elastic.co/guide/en/kibana/8.15/index.html

Following are the best practices for managing dashboards and saved objects within Arista Analytics, focusing on organization, maintainability, and upgrade compatibility:

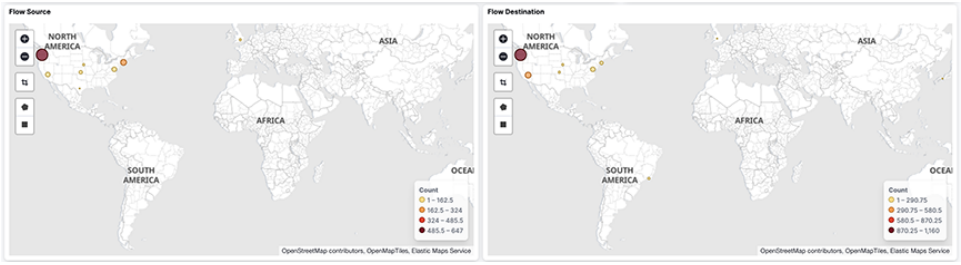
- **Consistent Naming Conventions**: Employ a naming convention that aligns with your environment.
  - Use prefixes to categorize dashboard content (for example, "ARISTA").
  - Include descriptive terms in the dashboard name to specify its type.
  - It improves organization and simplifies selection.
- **Simplified Management**: Consistent naming allows for easier individual selection and bulk operations.
  - Exporting dashboards based on their type facilitates tracking and management of modifications.
- **Upgrade Compatibility**: Build dashboards using custom visualizations and searches created for your environment.
  - Avoid relying on default objects, which might change during upgrades, potentially breaking your dashboards.

In summary, the best practices advocate for a structured and organized approach to dashboard management, ensuring maintainability, traceability, and resilience to system upgrades.

1.2.8 Geographic Location

- **GeoIP Database:** Arista Analytics uses the ***MaxMind GeoIP*** database to associate public network IP addresses with geographic locations.
- **Map Visualization:** This association displays a heat map on the **sFlow**® dashboard.
- **Geographic Filtering:** It filters the traffic shown on the map by selecting specific regions:
  - **Square Tool:** Draw a square to select a rectangular area.
  - **Polygon Tool:** Draw an irregular shape to select a more complex region.
  - **Zoom and Detail:** Selecting a region will zoom in on that area and provide more detailed information about the traffic flowing to or from it.

Figure 1-9: Geographic Flow Source and Destination



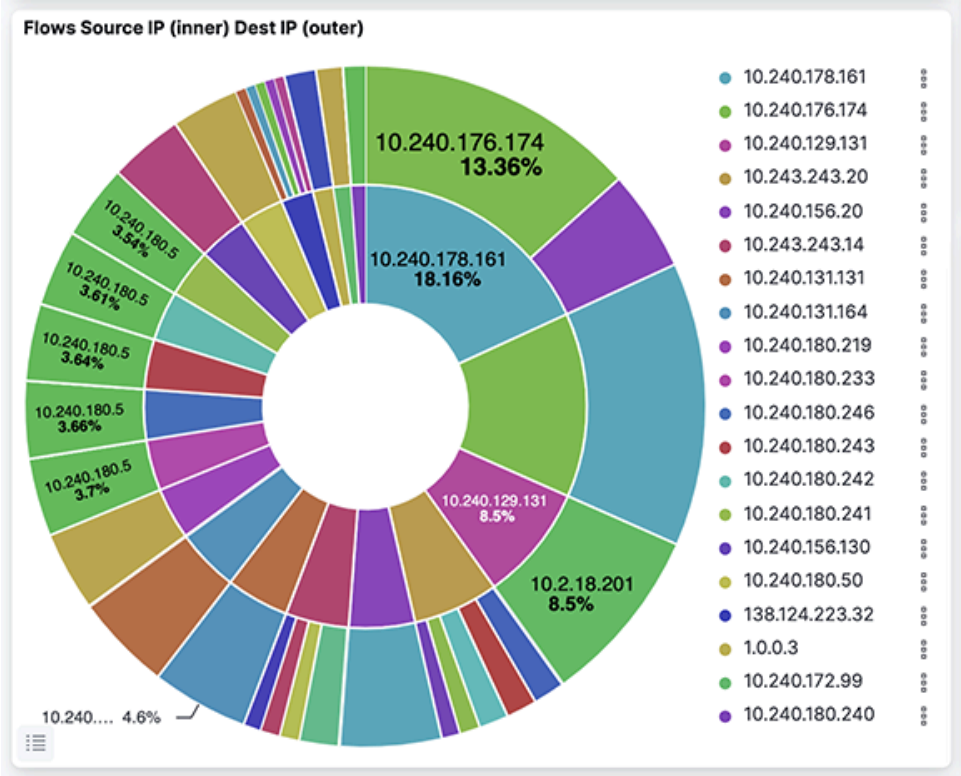
This process enables users to visually analyze network traffic patterns based on geographic location and focus on specific areas of interest for deeper investigation.

1.3 IP Addresses

This section describes identifying traffic transmitted or received by the source or destination IP address.

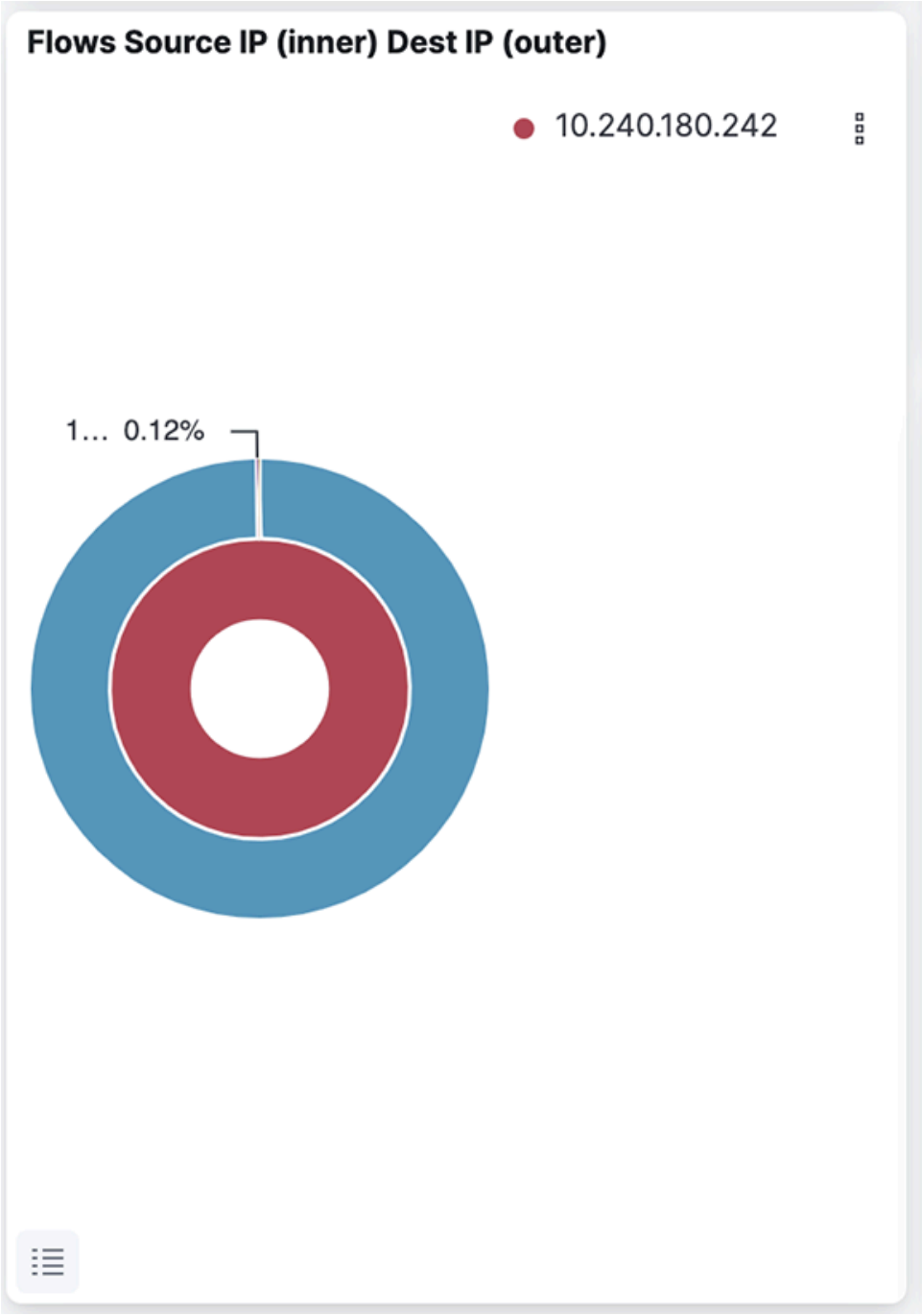
1.3.1 Source and Destination Addresses

Figure 1-10: Identifying Source and Destination IP Addresses



Click an IP address, then click the **Magnifying Glass** icon (+) to pin the address to the dashboard.

Figure 1-11: Filtering Results by IP Address



The selected IP address is added to the filters on the dashboard.

Each dashboard has a bar chart depicting traffic on the y-axis and time on the x-axis. To add a time filter, click and drag an area in the **All Flows Over Time** bar chart.

1.4 Interfaces Sending or Receiving Traffic

To identify specific interfaces that are sending or receiving traffic, select the following features:

- DMF Top Filter interfaces

- Production interfaces

Figure 1-12: DMF Filter Interfaces

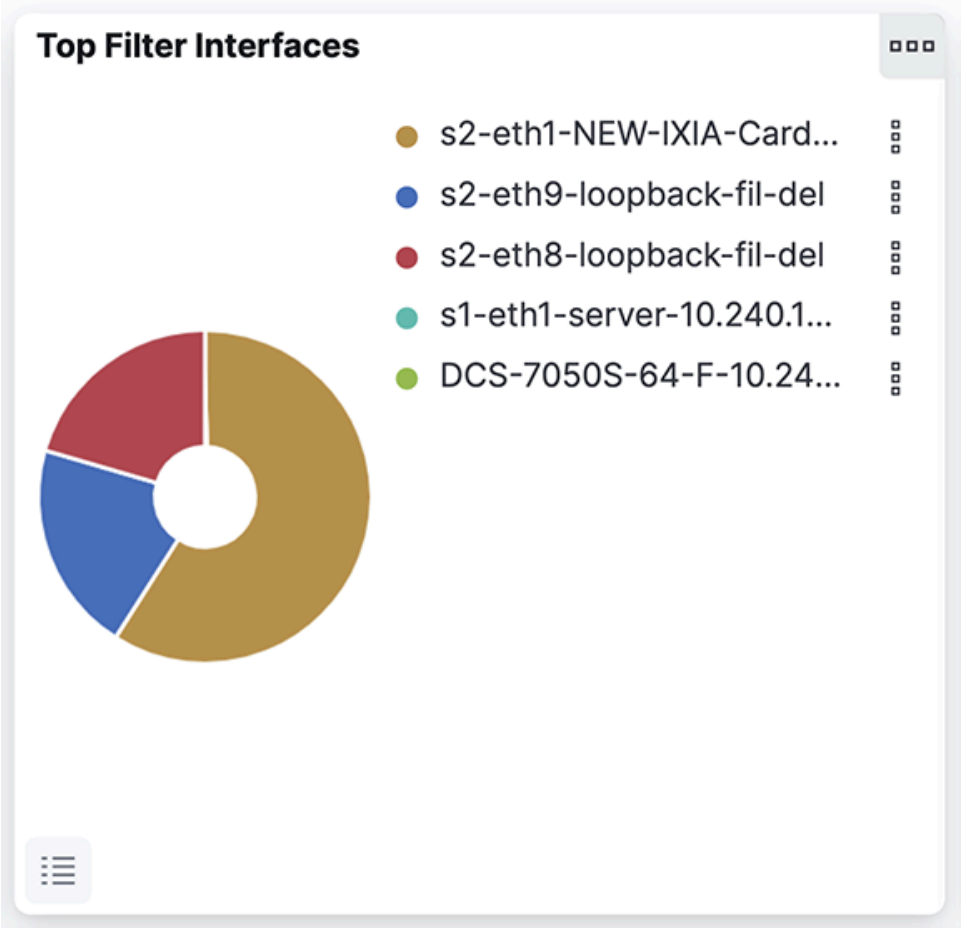
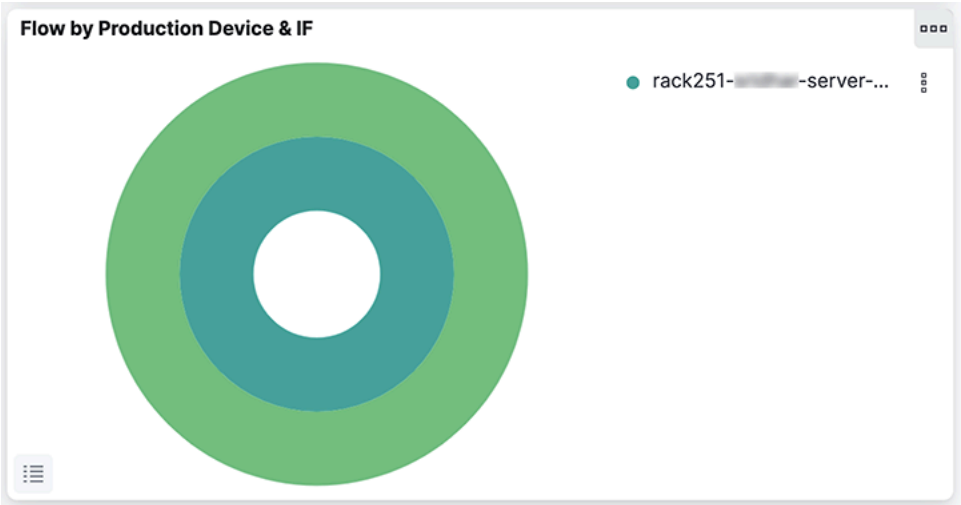


Figure 1-13: sFlow® > Flow by Production Device & IF



This information derives from the LLDP/CDP exchange between the production and DANZ Monitoring Fabric switches.

1.5 Filter Interface Information

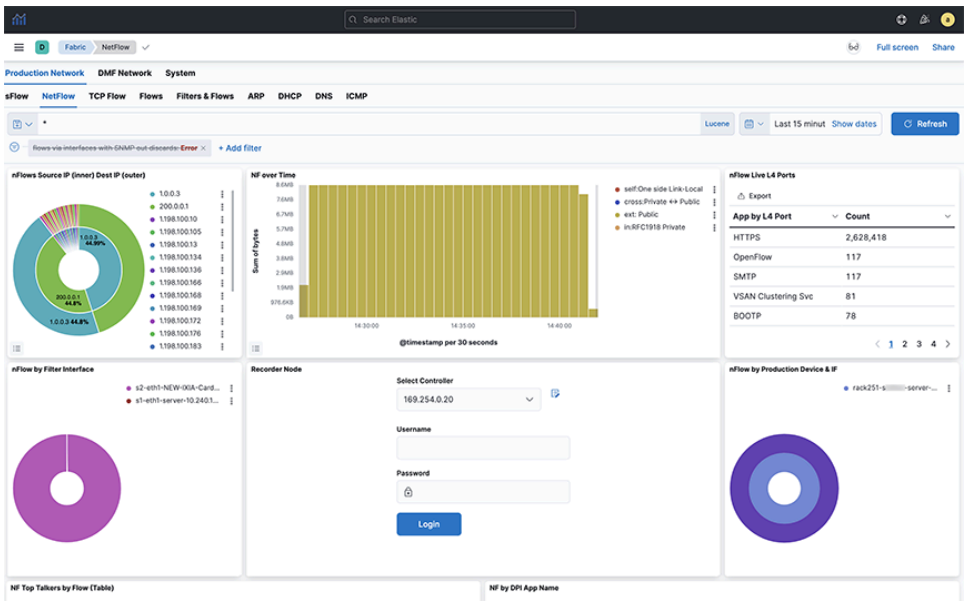
Add the filter interface name to the **NetFlow** dashboard to see hop-by-hop forwarding of flows for NetFlow traffic coming from the DMF Service Node for a specific flow. Arista Analytics then shows the filter interface name associated with that flow. It allows the network administrators to visualize the path a particular flow took through the network. If a flow goes through multiple hops, the dashboard would ideally display multiple "filter interface names," clearly indicating the sequence of interfaces.

1.5.1 Displaying Filter Interface Names

The **nFlow by Filter Interface** window on the **NetFlow** dashboard, shown later, can display the filter interface name where traffic is coming in for the NetFlow service. To display this information, enable the records-per-interface option in the NetFlow managed service configuration on the DANZ Monitoring Fabric Controller using the commands shown in the following example.

```
controller(config)# managed-service netflow-managed-service
controller(config-managed-srv)# service-action netflow netflow-delivery-int
controller(config-managed-srv-netflow)# collector 10.8.39.101 udp-port 2055 mtu 1500 records-per-i
nterface
```

Figure 1-14: Production Network > NetFlow Dashboard with Filter Interface Name



NetFlow Managed Service Records-per-interface Option

The following example displays the ***running-config*** for this configuration.

```
! managed-service
managed-service netflow-managed-service
 service-interface switch 00:00:4c:76:25:f5:4b:80 ethernet4/3:4
 !
 service-action netflow netflow-delivery-int
  collector 10.8.39.101 udp-port 2055 mtu 1500 records-per-interface
controller(config)# sh running-config bigtap policy netflow-policy
! policy
policy netflow-policy
 action forward
```

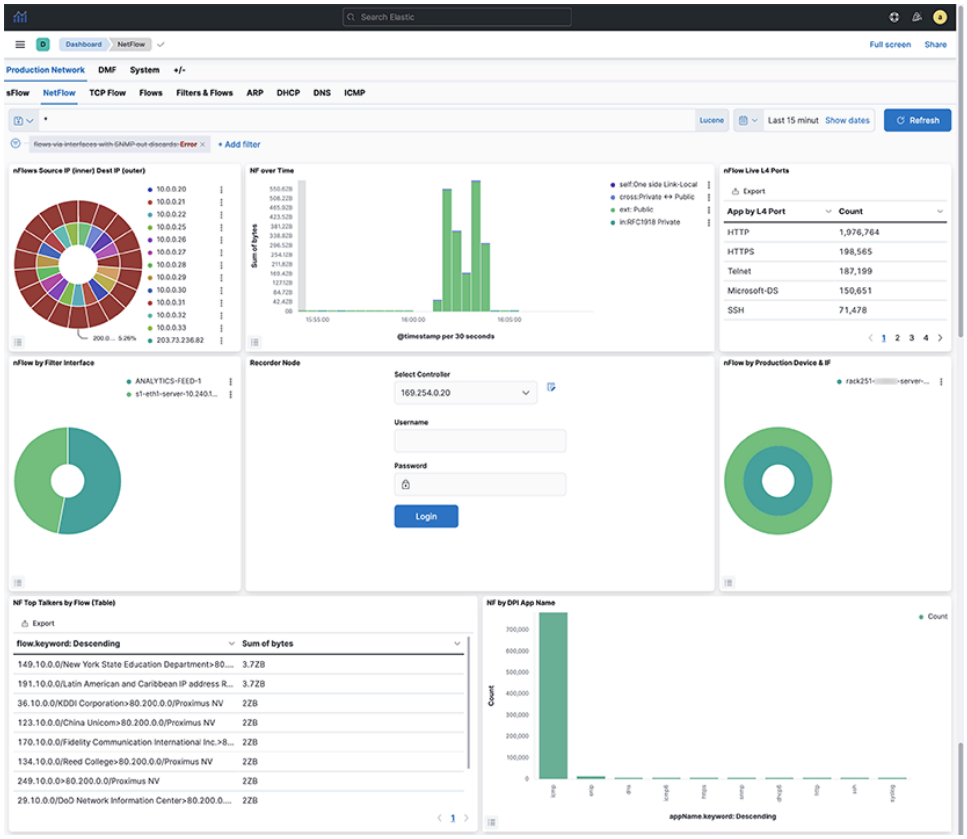
```
filter-interface filter-int-eth5
use-managed-service netflow-managed-service sequence 1 use-service-delivery
1 match any
```

After enabling this option, the **nFlow by Filter Interface** window, shown earlier, displays the filter interface identified in the policy that uses the NetFlow managed service.

The production device port connected to the filter interface sends LLDP messages; Arista Analytics also displays the production switch name and the production interface name attached to the filter interface in the **nFlow by Production Switch & IF** window.

In the example later, ***wan-tap-1*** displays in the **nFlow by Filter Interface** window. The production device N1524-WAN and the interface ***Gi1/0/1***, connected to filter interface ***wan-tap-1***, are displayed in the **nFlow by Production Switch & IF** window.

Figure 1-15: Production Network > NetFlow Dashboard with Filter Interface Name



1.6 WAN Link Identification

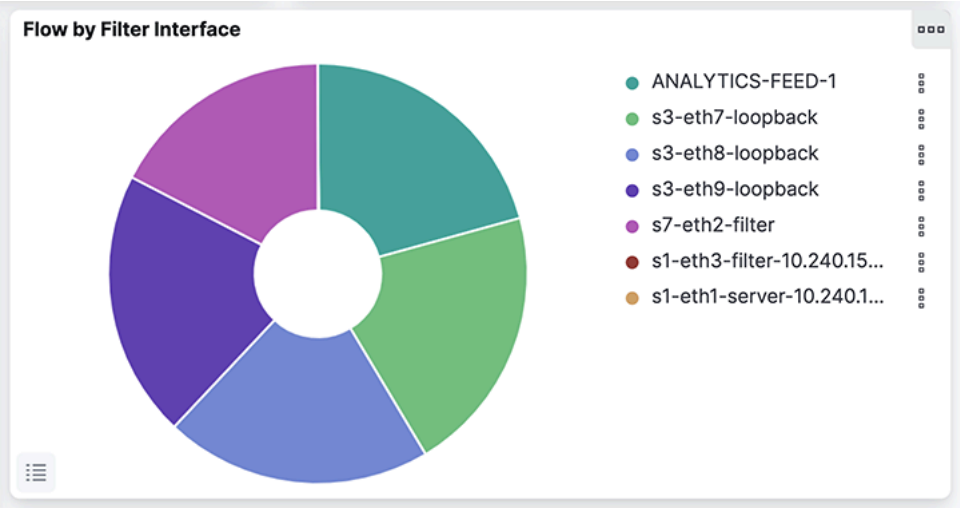
Use your knowledge of DMF filters or delivery interface names to monitor traffic to or from specific interfaces. DMF WAN interface names identified with a standard string, such as **wan**, can monitor the utilization of WAN links by reference to the DMF filter interface names.

To identify a WAN link or device that is approaching full utilization, complete the following steps:

1. Select **sFlow**.

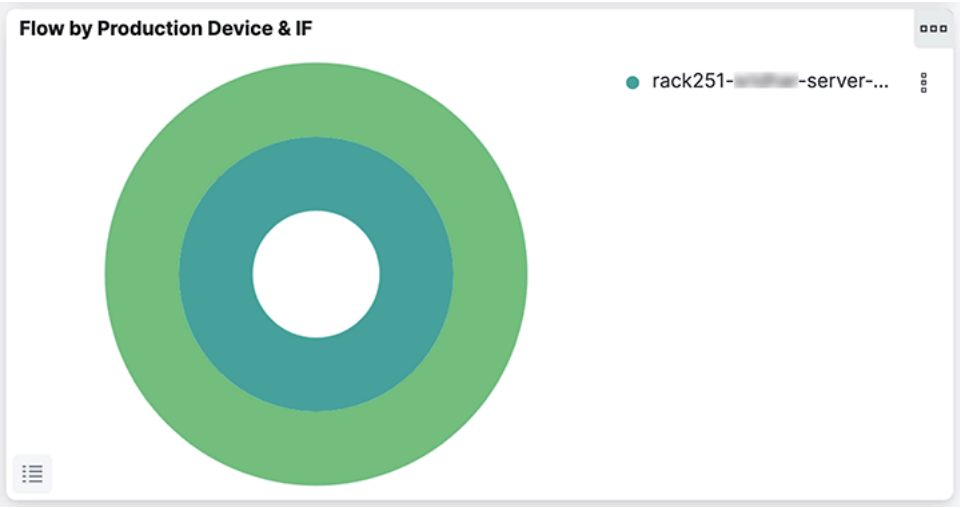
2. Refer to the **Flow by Filter Interface** visualization.

Figure 1-16: Flow by Filter Interface



This visualization displays the utilization for each DMF filter interface. To compare this to the traffic from the production interfaces (SPAN or Tap), use the **Flow by Production Device & IF** visualization.

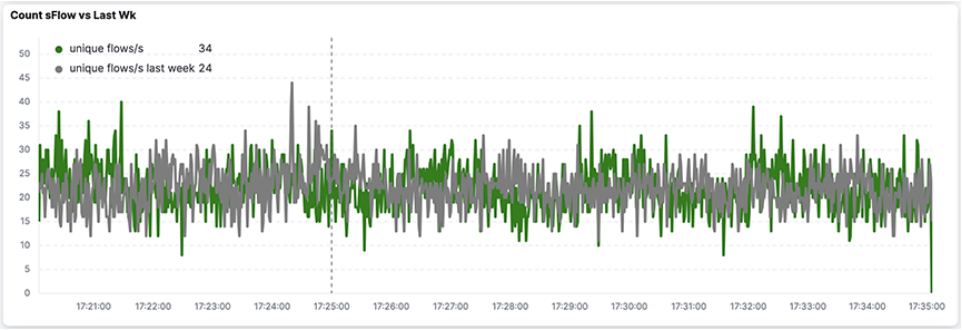
Figure 1-17: Flow by Production Device & IF



3. Select the Filter Interfaces corresponding to the WAN link.

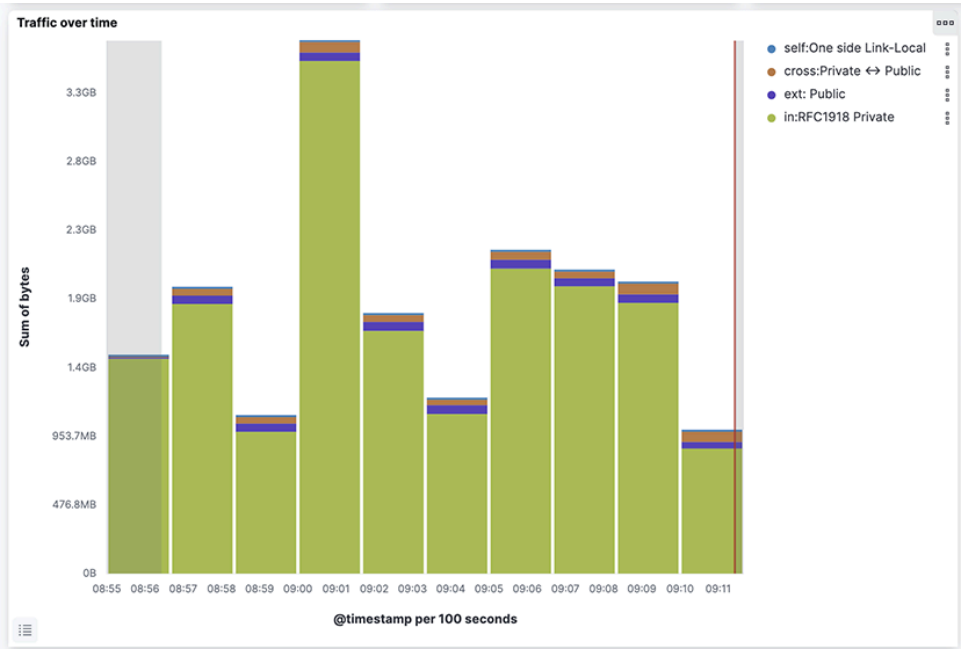
Refer to the **Count sFlow vs Last Wk** visualization to determine if any significant change in utilization has occurred.

Figure 1-18: Count sFlow vs Last Wk



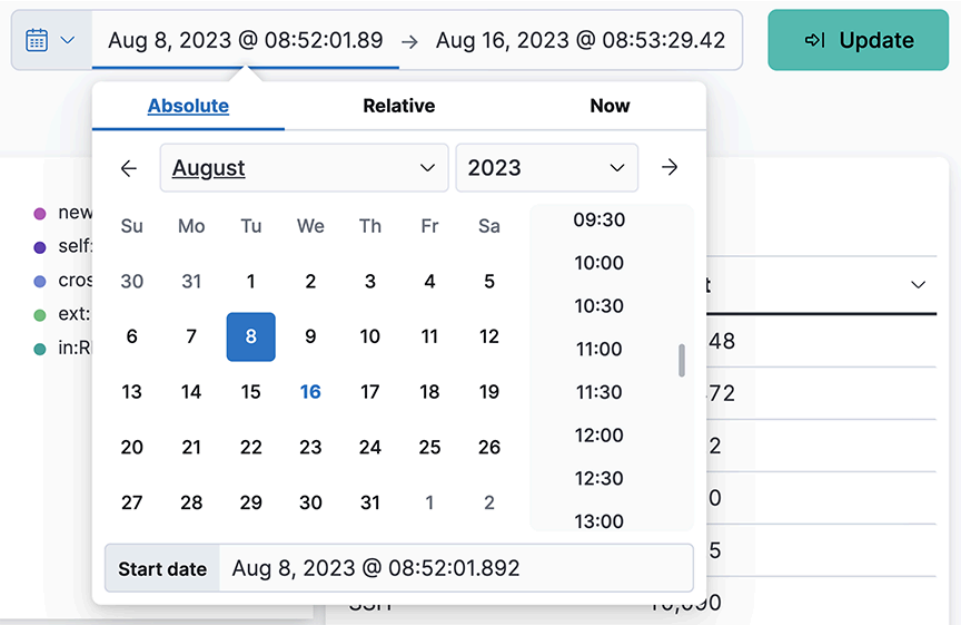
Use the **Traffic over Time** visualization to focus on peak and non-peak utilization periods. Drag the cursor horizontally over a peak utilization period, and the display is updated to zoom in on those events.

Figure 1-19: Traffic Over Time



4. Use the **Time Range** configuration to analyze traffic over a month for a more complete characterization.

Figure 1-20: Expanding Time Period Using the Time Range



Chapter 2

Production

This chapter describes the dashboards provided on the Production Network tab, which shows traffic and events on the production network interfaces connected to the DANZ Monitoring Fabric. This section includes the following sections:

2.1 sFlow®

Select the **sFlow*** from the left-hsnd navigation bar in **Production** tab, and It summarizes information from the sFlow messages sent to the Arista Analytics server from the DANZ Monitoring Fabric Controller or other sFlow agents. This dashboard provides the following panels:

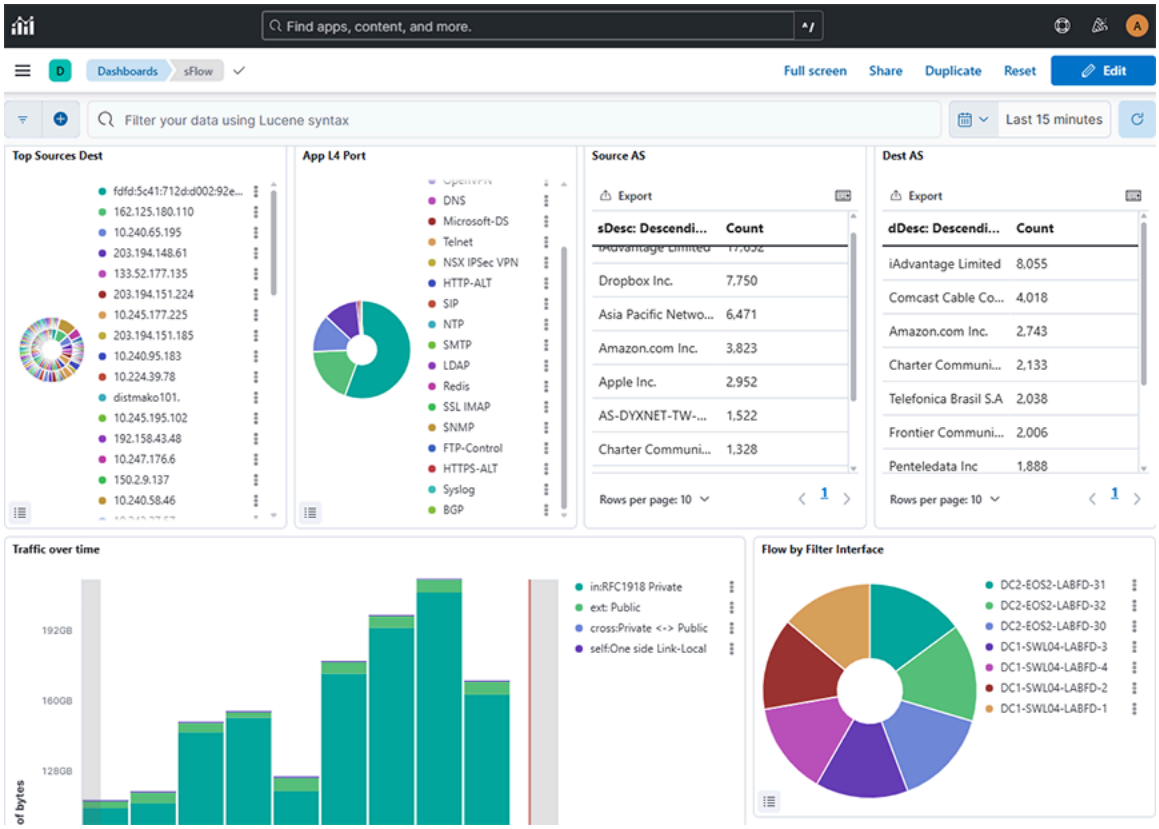
- **Top Sources Dest**: Configures a real-time network traffic monitoring dashboard using sFlow data, displaying flows over a selected time range grouped by source (***sHost.keyword***) and destination (***dHost.keyword***) IPs/hosts, sorted descendingly to highlight top talkers and recipients for rapid traffic pattern analysis.
- **App L4 Port**: Configures a real-time sFlow traffic dashboard displaying flows over a selected time range grouped by L4 application protocols (***l4App.keyword***) and sorted descendingly to highlight the most active applications for quick protocol-level traffic analysis.
- **Source AS**: Displays a real-time tabular view of sFlow traffic data, showing the top source descriptions (***sDesc***) by flow count in descending order to highlight the most active network entities over a selected time range.
- **Dest AS**: Displays a real-time tabular view of sFlow traffic data, showing the top source descriptions (***dDesc***) by flow count in descending order to highlight the most active network entities over a selected time range.
- **Traffic over Time**: Configures a time-based stacked bar chart using sFlow data, showing the sum of upsampled byte counts over a selected time range, categorized by traffic type filters to analyze traffic distribution across private, public, and link-local networks.

* sFlow® is a registered trademark of the Inmon Corp.

- **Flow by Filter Interface**: Displays a donut chart using sFlow data to show the count of flow records over a selected time range, grouped by BT interface name (***BTifName***) in descending order to highlight the most active interfaces.
- **Flow by Production Device & IF**: Displays a nested donut chart using sFlow data over a selected time range, grouping flow record counts by device and device port in descending order to identify the most active devices and their respective ports in the network.

Selecting **Dashboards** → **sFlow** displays network traffic on the **Analytic Node**.

Figure 2-1: Production > sFlow Dashboard- Top

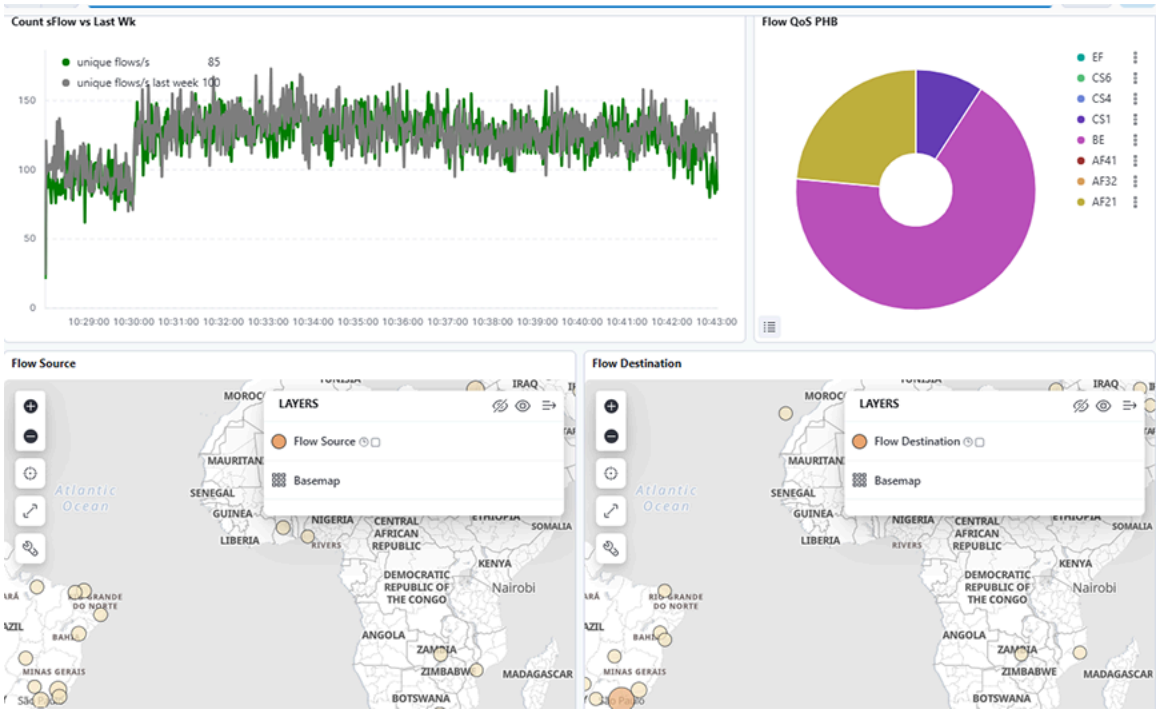


The middle dashboard include:

- **Count sFlow vs. Last Wk**: Tracks the rate of unique network flows per second using sFlow data over a selected time range. It compares it with the same metric from the previous week for anomaly detection and traffic trend analysis.
- **Flow QoS PHB**: Configures a real-time traffic classification dashboard using sFlow data, displaying the count of flows over a selected time range grouped by Differentiated Services Code Point (DSCP) values (***phb.keyword***), helping identify traffic prioritization and QoS usage patterns.
- **Flow Source**: Displays global network traffic sources using sFlow data over a selected time range, mapped geographically to highlight the density and distribution of flow sources across regions.

- **Flow Destination**: Displays global network traffic destinations using sFlow data over a selected time range, mapped geographically to highlight the density and distribution of flow destinations across regions.

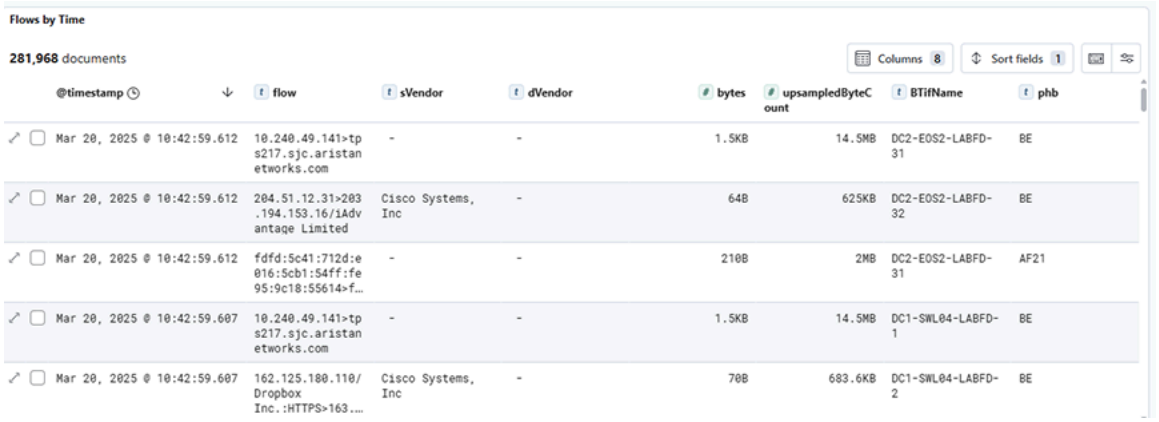
Figure 2-2: Production > sFlow Dashboard- Middle



The lower dashboard includes:

- **Flows by Time**: Displays detailed raw sFlow traffic records over a selected time range, showing individual flow entries along with associated metadata such as source/destination vendors, byte counts, interface names, and PHB values for in-depth traffic inspection.

Figure 2-3: Production > sFlow Dashboard- Bottom



2.1.1 sFlow® Monitoring of MPLS and GRE Tunnels

Arista Analytics can process sFlow® records containing IP packets encapsulated in additional protocol headers:

1. IPv4 and IPv6 packets encapsulated in one or more MPLS headers.
2. IPv4 and IPv6 packets encapsulated in one or more GRE headers.
3. Combination of (1) and (2).
4. VXLAN encapsulated in one or more GRE headers

The following table shows the supported combinations of headers.

Table 1: Ethernet Support

1	Ethernet	MPLS(one or more)	IPv4/IPv6	TCP/UDP		
2	Ethernet	IPv4/IPv6	GRE + IPv4/ IPv6(one or more)	TCP/UDP		
3	Ethernet	MPLS(one or more)	IPv4/IPv6	GRE + IPv4/ IPv6(one or more)	TCP/UDP	
4	Ethernet	IPv4/IPv6	GRE + IPv4/ IPv6(one or more)	UDP	VxLAN	Ethernet

This feature enhances the visibility of tunnelled traffic by documenting the content of packet headers in Analytics' Elasticsearch database. It does not affect the current formatting of Elasticsearch documents containing VXLAN headers not encapsulated by any other tunnel header.

Elasticsearch Documents

It displays each encapsulated header in a JSON object whose name has an ‘e’ prefix followed by an integer corresponding to the position of the header in the packet, 1 being the outermost tunnel header. It displays the innermost header in a JSON object named *innerPkt*. The following is an example of such a JSON object:

```
"e1": {
  "encapType": "MPLS",
  "label": 90,
  "ttl": 10
 },
 "e2": {
  "encapType": "MPLS",
  "label": 91,
  "ttl": 9
 },
 "e3": {
  "encapType": "IPv4",
  "sIp": "10.99.230.30",
  "dIp": "10.51.49.227",
  "proto": 47
 },
 "e4": {
  "encapType": "GRE",
  "proto": 34525
 },{
 "sMac": "e2:d8:dd:e3:6a:57",
 "dMac": "84:3d:9c:be:a7:1a",

 "e5": {
  "encapType": "IPv6",
  "sIp": "4cdd:2cde:2ead:248e:1e9b:c3dc:9e3f:3d2b",
  "dIp": "9f0a:a91a:7e79:31e:78c8:e47b:1cd2:e783",
  "proto": 47
 }.
```

```
  "e6": {
   "encapType": "GRE",
   "proto": 2048
  },
  "e7": {
   "encapType": "IPv4",
   "sIp": "10.218.163.106",
   "dIp": "10.79.78.163",
   "proto": 17
  },
  "e8" {
   "encapType": "UDP",
   "sP": 9999,
   "dP": 4739
  },
  "e9": {
   "encapType": "VXLAN",
   "vni": 8888
  },
  "innerPkt": {
   "sMac": "16:af:16:2e:83:0c",
   "dMac": "e6:32:ff:5e:50:3f",
   "sIp": "10.44.66.146",
   "dIp": "10.218.200.227",
   "proto": 6,
   "sP": 7777,
   "dP": 443
  }
 }
```

Each header displays a different set of information.

For MPLS:

- *encapType*: Always set to **MPLS**
- *label*: MPLS label number
- *ttl*: MPLS TTL

For GRE:

- *encapType*: Always set to **GRE**
- *proto*: Ethernet type
- *key*: GRE key, if present
- *seqNum*: GRE sequence number, if present

For VXLAN:

- *encapType*: Always set to **VXLAN**
- *vni*: VNI number

For IP headers that follow GRE headers:

- *encapType*: **IPv4** or **IPv6**
- *sIp*: Source IP address
- *dIp*: Destination IP address
- *proto*: IP protocol number

For UDP header preceding an encapsulated VXLAN header:

- *encapType*: **UDP**
- *sP*: Source port

- *dP*: Destination port

Troubleshooting

If sFlow documents do not appear in ElasticSearch, Arista Networks recommends creating a support bundle and contacting Arista TAC.

Limitations

- Since MPLS headers do not include any information about the encapsulated packets, processing of MPLS encapsulating any header other than IPv4 or IPv6 will result in undefined behaviour.
- It displays the IP addresses of headers that are outer-tunnelled the sFlow dashboard, but nothing beyond (e.g., UDP/TCP port number in inner headers).
- This feature does not support the PseudoWire control word.
- This feature supports up to 4 MPLS labels displayed on the dashboard. Packets with 5 or more MPLS labels will not appear on the dashboard, but they are present in Elasticsearch.

2.2 NetFlow

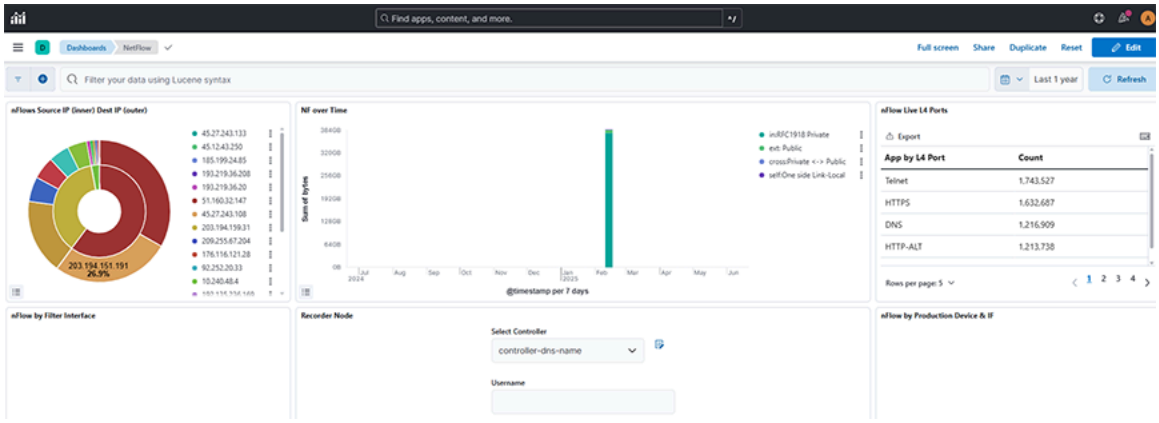
Selecting **Production** → **NetFlow** displays data transmitted on the **Analytic Node**.

The dashboard displays:

- **nFlows Source IP (inner) Dest IP (outer)**: Displays NetFlow data over a selected time range, showing the count of flows grouped by source (slp) and destination (dlp) IP addresses in descending order to identify the most active endpoints.
- **NF over Time**: Displays the sum of bytes over a selected time range, with time-based aggregation on the X-axis and filtered series split by traffic direction tags (for example, internal, external, cross, self) for comparative flow analysis.
- **nFlow Live L4 Ports**: Displays the count of NetFlow records over a selected time range, grouped by the ***l4App.keyword*** field using a row-based split to categorize traffic based on Layer 4 application types.
- **nFlow by Filter Interface**: Displays NetFlow data as a pie chart over a selected time range, counting flow records grouped by ***BTifName*** (interface name) in descending order to highlight the top 50 interfaces by activity.
- **Recorder Node**

- **nFlow by Production Device & IF**: Displays NetFlow data as a nested pie chart over a selected time range, showing the count of flow records grouped first by device name and then by device port, sorted in descending order to highlight the most active sources and interfaces.

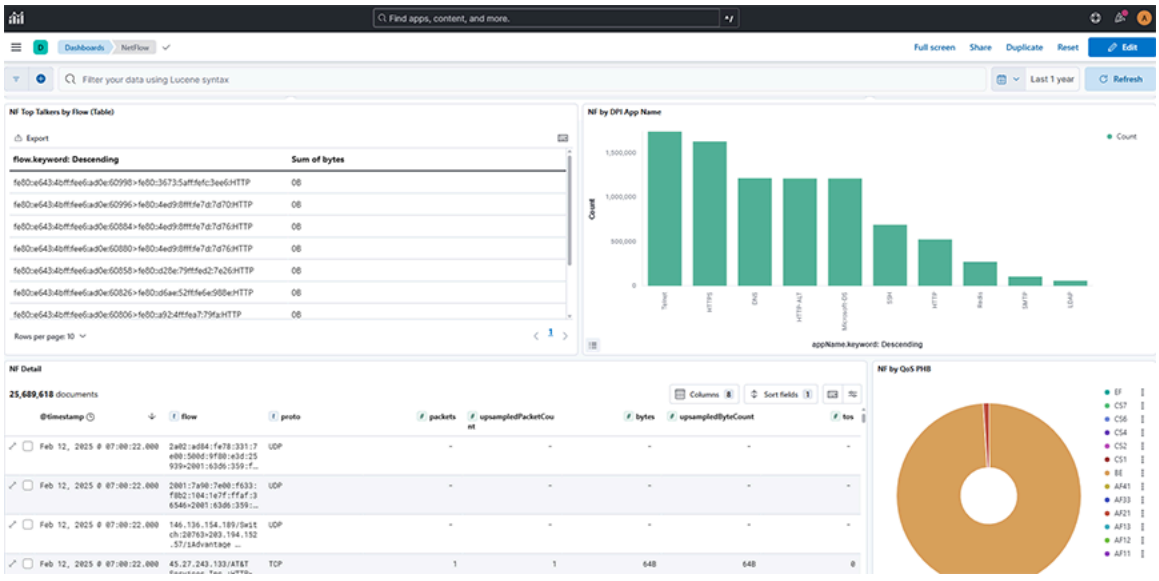
Figure 2-4: Production > NetFlow Dashboard - Top



The bottom dashboard includes:

- **NF Top Talkers by Flow (Table)**: Displays NetFlow records in a table format over a selected time range, showing flow identifiers sorted in descending order to highlight the top talkers by flows.
- **NF by DPI App Name**: Displays a count of NetFlow records over a selected time range, grouped by application name (***appName.keyword***) to identify the most active applications in the network traffic in descending order.
- **NF Detail**: Displays detailed NetFlow records over a selected time range, showing individual flow entries and associated metadata such as source/destination IPs, protocol, packet, and byte counts, enabling granular network traffic inspection.
- **NF by Qos PHB**: Displays a count of NetFlow records over a selected time range, grouped by Per-Hop Behavior (***phb.keyword***) in descending alphabetical order to analyze differentiated services traffic classification.

Figure 2-5: Production > NetFlow Dashboard- Bottom



2.3 TCP Flow

Selecting **Production** → **TCPFlow** displays data transmitted on the **Analytic Node**.

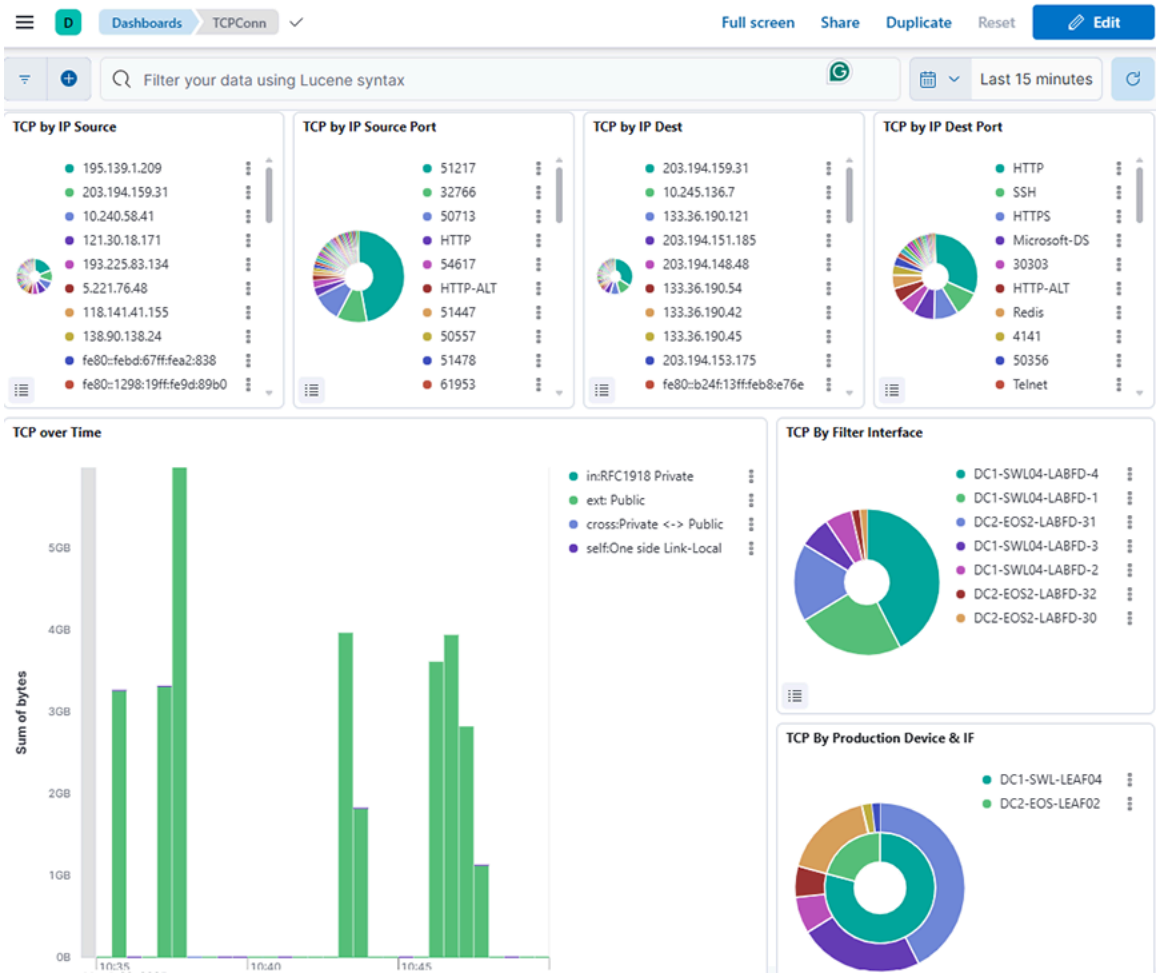
The information on the TCP Flow dashboard depends on TCP handshake signals and deduplicates. The switch description is specified in the **Description attribute** of each switch, configured on the DANZ Monitoring Fabric Controller.

The dashboard displays:

- **TCP by IP Source**: Displays the distribution of TCP connections over a selected time range, grouped by Source IP address (sip) and ordered by count in descending order.
- **TCP by IP Source Port**: Displays the distribution of TCP connections over a selected time range, grouped by source port (***sP.keyword***) and sorted by connection count in descending order.
- **TCP by IP Dest**: Displays the distribution of TCP connections over a selected time range, grouped by destination IP address (dlp) and ordered by count in descending order.
- **TCP by IP Dest Port**: Displays the distribution of TCP connections over a selected time range, grouped by destination port (***dP.keyword***) and sorted by connection count in descending order.
- **TCP over Time**: Displays the total bytes of TCP traffic over a selected time range, grouped by timestamp and categorized by traffic direction using tag-based filters (for example, private, public, and cross-zone flows).
- **TCP by Filter Interface**: Displays the distribution of TCP connection counts over a selected time range, categorized by BT interface name using descending order of occurrence.

- **TCP By Production Device & IF**: Displays the count of TCP connections over a selected time range, grouped first by device name and then by device port, using a nested pie chart.

Figure 2-6: Production > TCPFlow Dashboard - Top

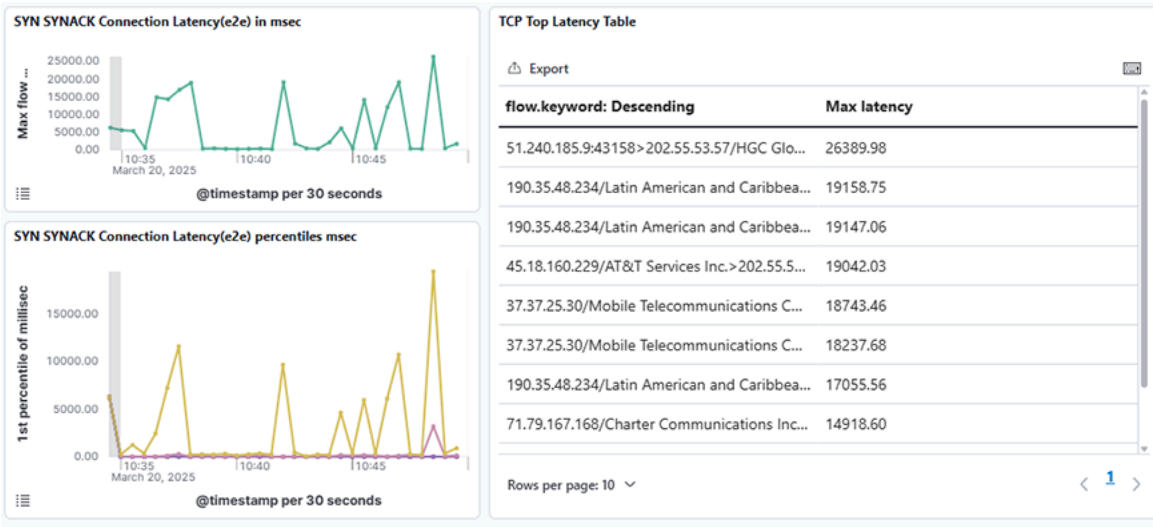


The middle dashboard includes:

- **SYN SYNACK Connection Latency(e2e) in msec**: Displays the maximum end-to-end SYN-SYNACK connection latency in milliseconds over a selected time range using a time-based line graph.
- **SYN SYNACK Connection Latency(e2e) percentiles**: Displays the SYN-SYNACK end-to-end connection latency across multiple percentiles (1st, 5th, 25th, 50th, 75th, and 95th) over a selected time range using a time-series line graph.

- **TCP Top Latency Table**: Displays the top TCP flows sorted by their maximum observed latency over a selected time range.

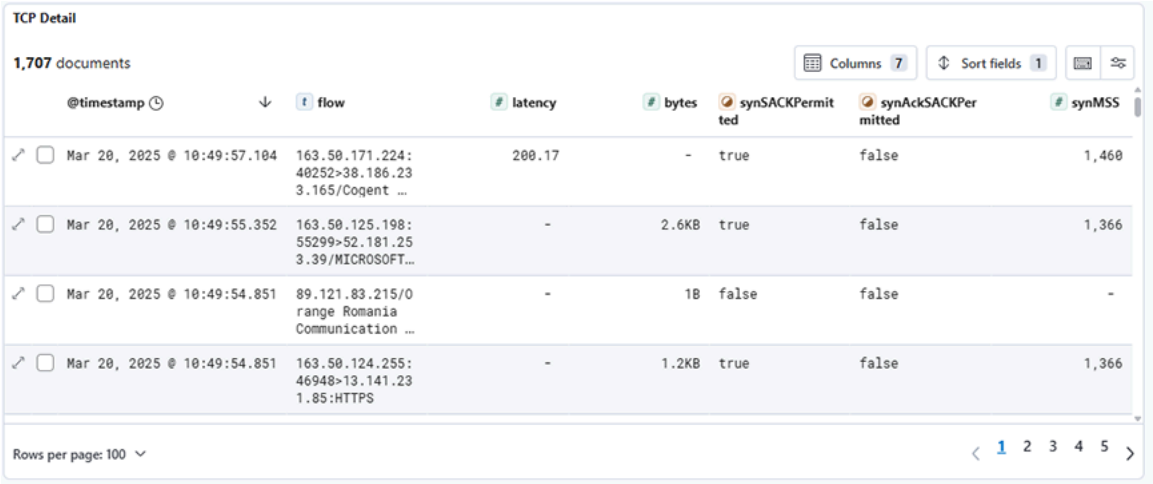
Figure 2-7: Production > TCPFlow Dashboard- Middle



The lower dashboard includes:

- **TCP Detail**: Displays detailed TCP connection records filtered by ***type:tcpconnection***, showing attributes such as latency, byte count, and SYN/SYNACK flags for each flow over a selected time range.

Figure 2-8: Production > TCPFlow Dashboard- Bottom



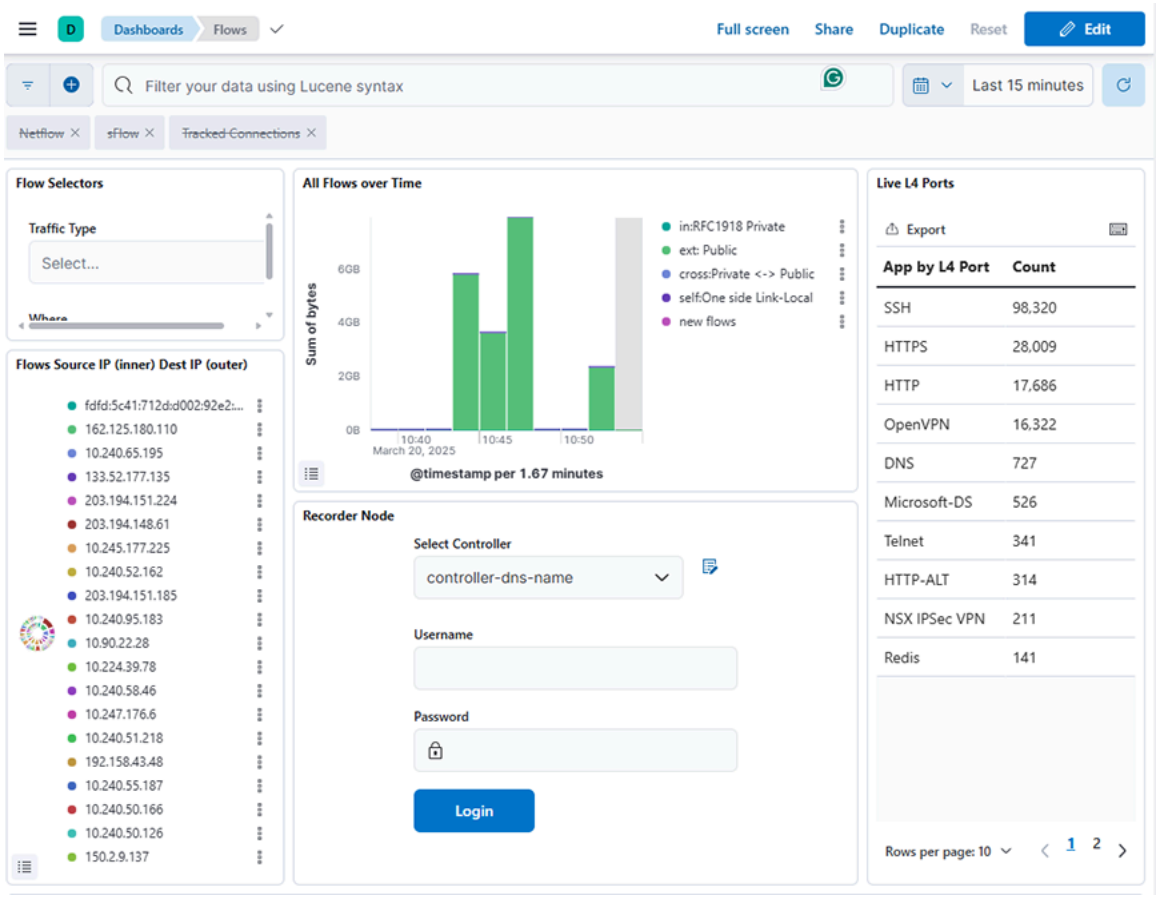
2.4 Flows

Selecting **Production** → **Flows** displays the network's data efficiency, high traffic monitoring and scalability on the **Analytic Node**.

The dashboard displays:

- **Flow Selector**: Allows network traffic data to be filtered using dynamic dropdowns for **Traffic Type** based on ***tags.keyword*** field, enabling users to interactively refine insights over a selected time range.
- **All Flows over Time**: Displays the sum of bytes for different traffic categories (for example, private, public, cross-zone, link-local, and new flows) over a selected time range, using a date histogram with 100-second intervals.
- **Live L4 Port**: Displays the count of flow records grouped by L4 application port over a selected time range using the ***l4App.keyword*** field for aggregation.
- **Flows Source IP (inner) Dest IP (outer)**: Displays the count of flows grouped first by source IP (sip) and then by destination IP (dip) over a selected time range.
- **Recorder Node**: Refer to DMF Recorder Node.
- **Flow Heatmap VPCs**: Displays a heatmap of the total bytes transferred between source VPCs and destination VPCs over a selected time range.

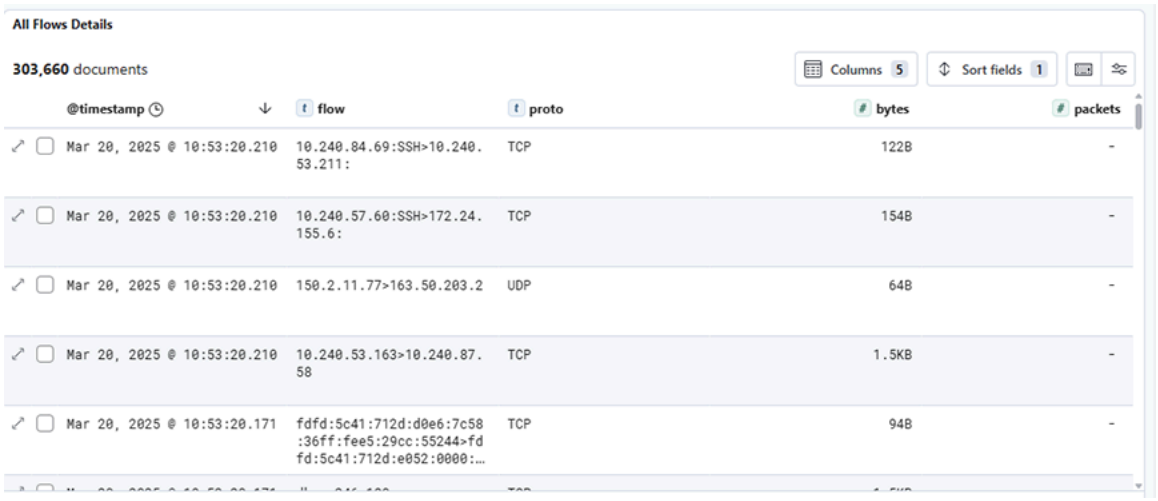
Figure 2-9: Production > Flows Dashboard - Top



The lower dashboard includes:

- **All Flows Details**: Displays detailed flow records filtered by flow type, including protocol, byte count, and packet count, over a selected time range.

Figure 2-10: Production > Flows Dashboard- Bottom



2.5 Filters & Flows

Selecting **Production** → **Filters & Flows** displays the selected MAC Address to access the **Analytic Node**.

The dashboard displays:

- **Top Filters**: Displays the sum of bytes over time, grouped by top ***BTIfName*** values, over a selected time range.
- **Filter Top Flows**: Displays the sum of bytes sent as flow traffic over time, split by ***BTIfName***, and each segment further splits by ***flow1.keyword*** (descending)., over a selected time range.

- **Filter Bottom Flows**: Displays the sum of bytes sent as flow traffic over time, split by ***BTIfName***, and each segment further splits by ***flow1.keyword*** (ascending)., over a selected time range.

Figure 2-11: Production > Filters & Flows Dashboard - Top

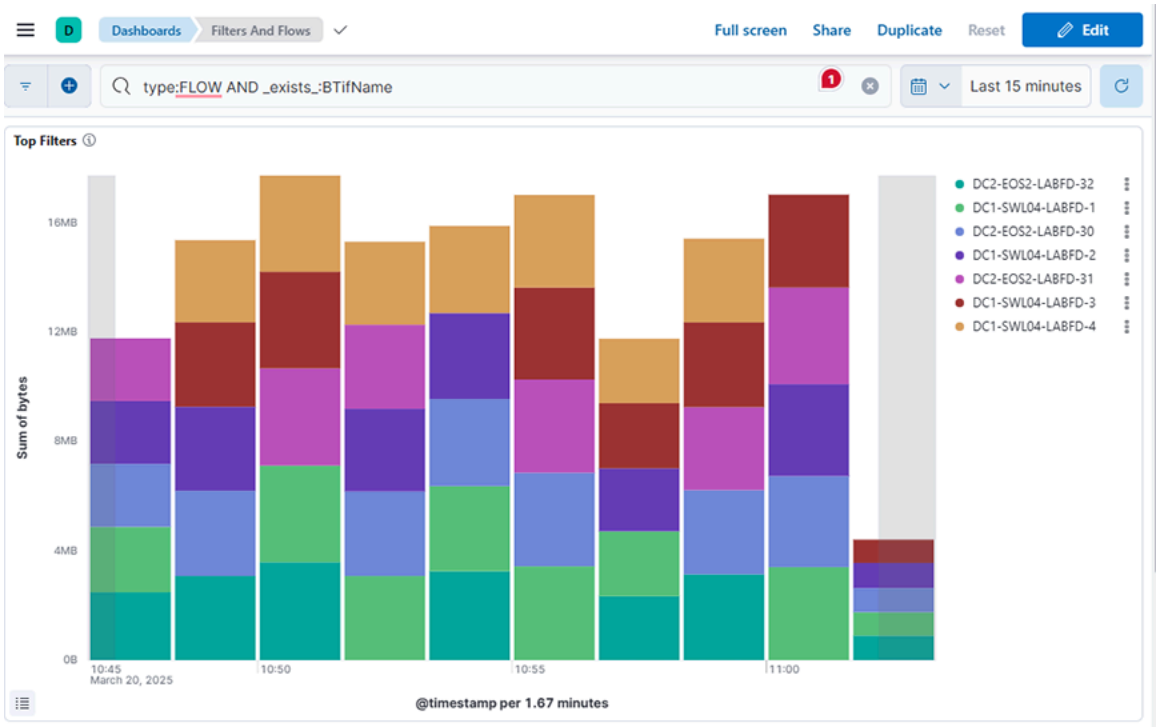
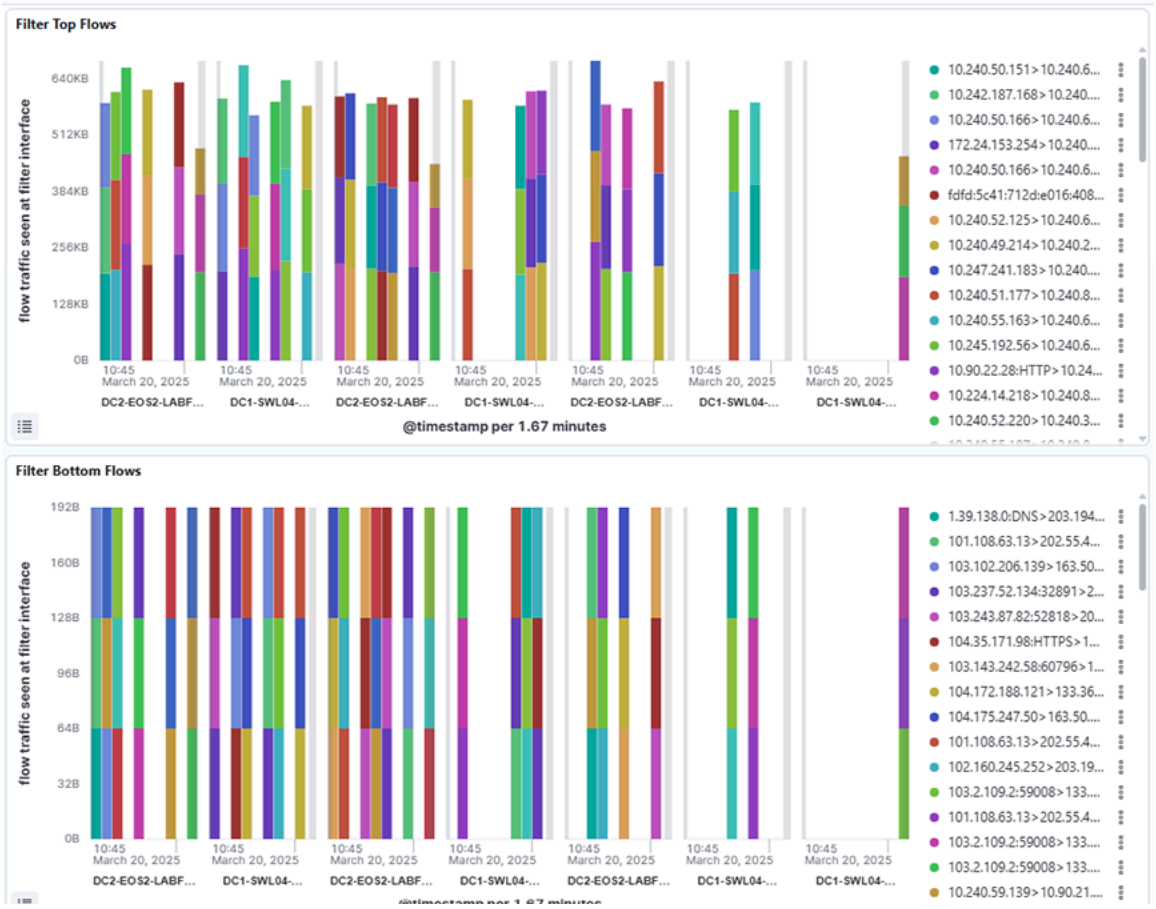


Figure 2-12: Production > Filters & Flows Dashboard - Bottom



2.6 ARP

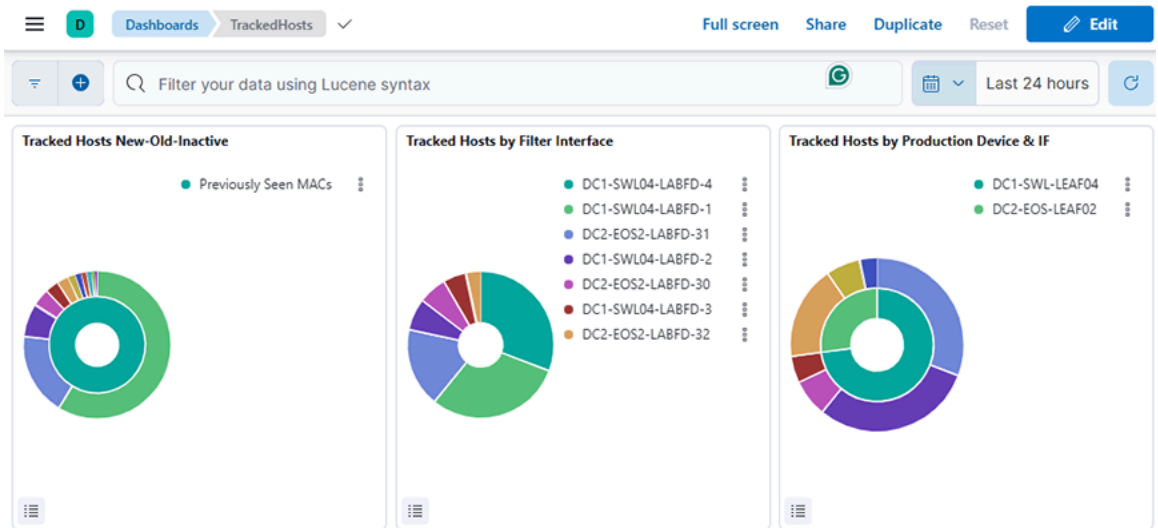
This data correlates with the tracked host feature on the DANZ Monitoring Fabric Controller. It shows all ARP data when you switch interface and production devices over time.

The **Production** → **ARP** displays:

- **Tracked Hosts New-Old-Inactive**: Displays the count of tracked hosts categorized as newly seen, previously seen, or inactive, grouped by OUI vendor over a selected time range.
- **Tracked Hosts by Filter Interface**: Displays the unique count of tracked MAC addresses grouped by filter interface over a selected time range.

- **Tracked Hosts by Production Device & IF**: Displays the unique count of tracked MAC addresses grouped by production device and interface over a selected time range.

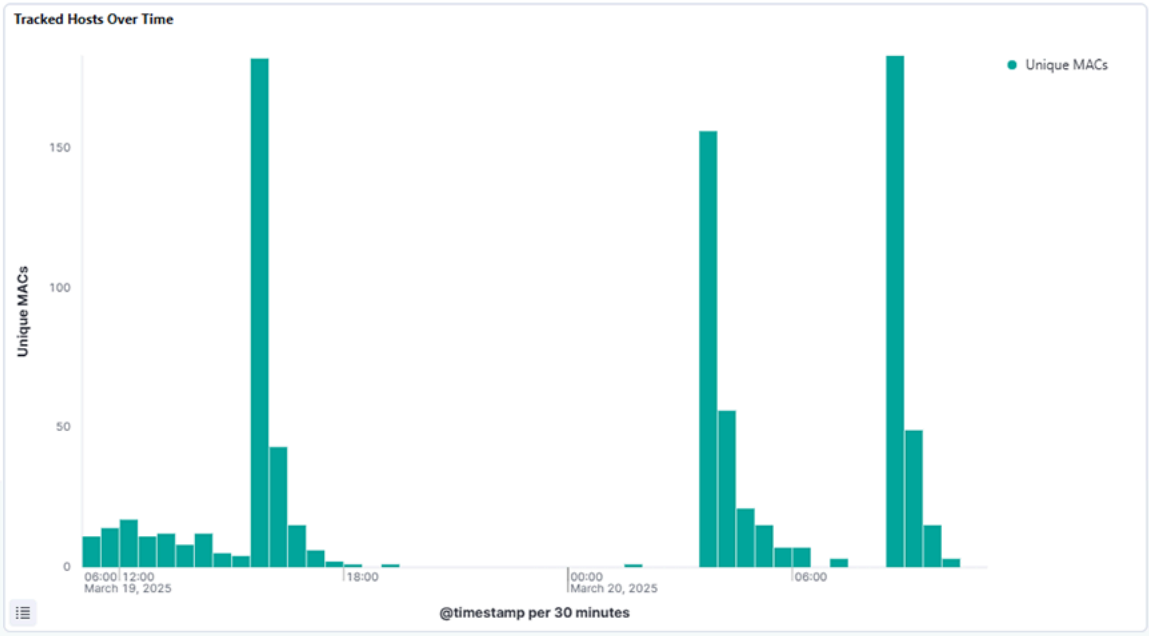
Figure 2-13: Production > ARP Dashboard - Top



The middle dashboard includes:

- **Tracked Hosts Over Time**: Displays the unique count of MAC addresses over a selected time range using a date histogram to track the number of distinct tracked hosts seen over time.

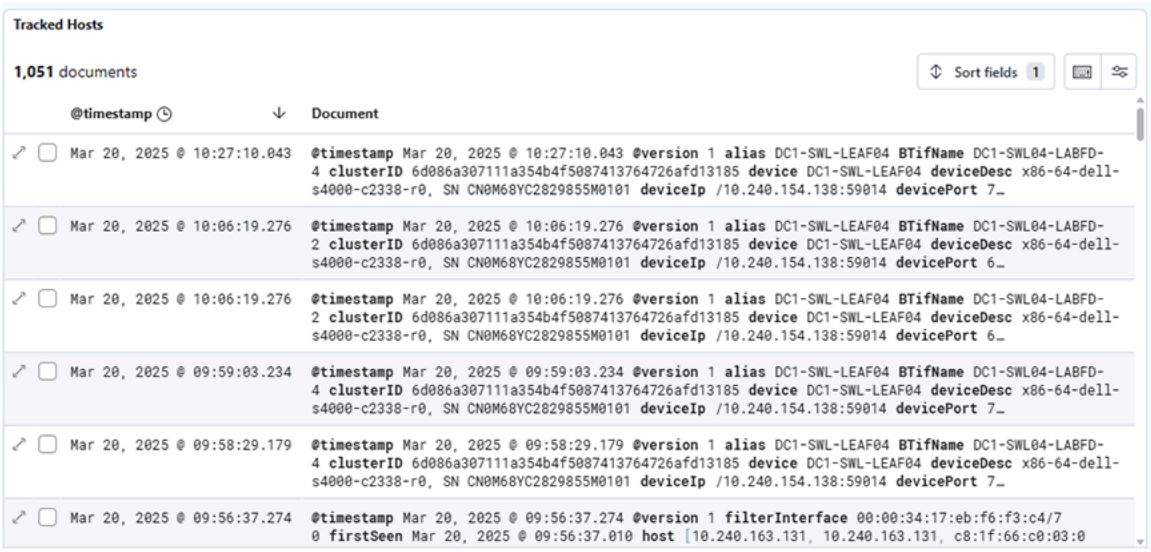
Figure 2-14: Production > ARP Dashboard- Middle



The lower dashboard includes:

- **Tracked Hosts**: Displays raw tracked host events over a selected time range, showing detailed logs with metadata such as timestamp, device, port, hostname, interface, and activity status.

Figure 2-15: Production > ARP Dashboard- Bottom



2.7 DHCP

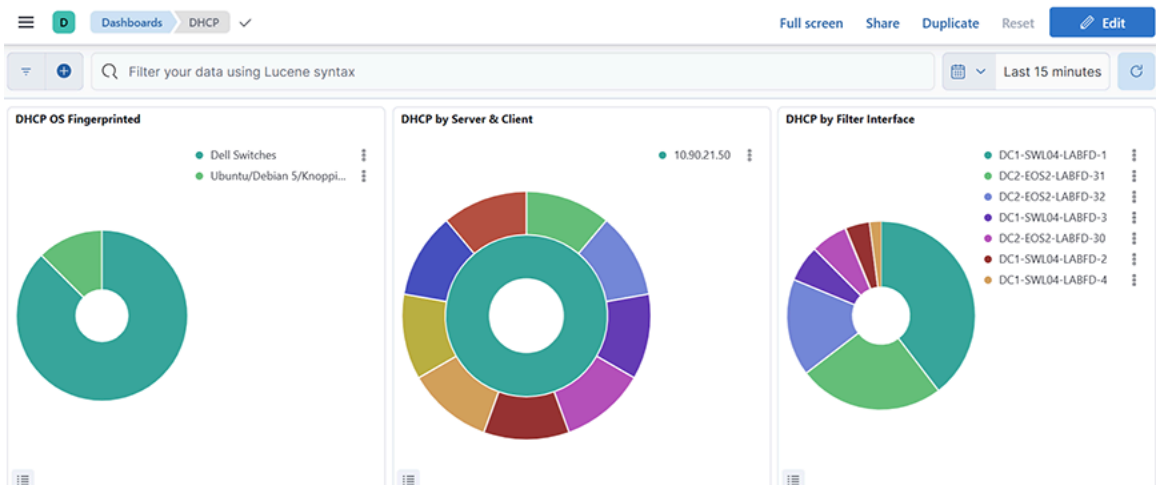
Selecting **Production** → **DHCP** analyzes DHCP activity on the **Analytic Node**.

The **DHCP Dashboard** summarizes information from analyzing DHCP activity and provides the following panels. The dashboard displays:

- **DHCP OS Fingerprinted**: Displays the distribution of fingerprinted operating systems based on the unique count of host MAC addresses over a selected time range
- **DHCP by Server & Client**: Displays the top server names and associated CNAMEs based on count over a selected time range.

- **DHCP by Filter Interface**: Displays the distribution of DHCP message activity by filter interface, based on count, over a selected time range.

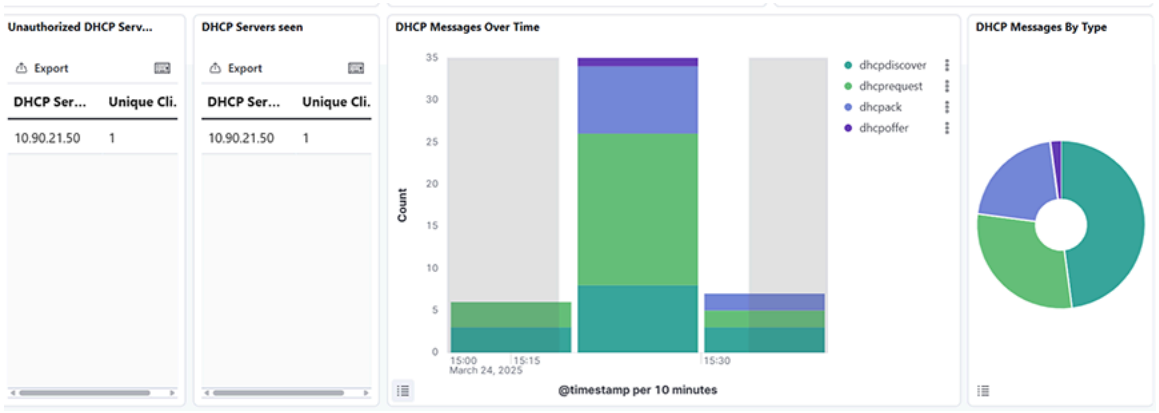
Figure 2-16: Production > DHCP Dashboard - Top



The middle dashboard includes:

- **Unauthorized DHCP Server**: Identifies unauthorized DHCP servers by displaying the unique number of clients they responded to with DHCP offers, excluding known valid servers, over a selected time range.
- **DHCP Server seen**: Displays all DHCP servers that sent offers and the count of unique clients they responded to, excluding ***0.0.0.0*** addresses, over a selected time range.
- **DHCP Messages Over Time**: Displays the distribution of DHCP message types (for example, discover, offer, request, ack) over a selected time range, using a date histogram split by message type.
- **DHCP Messages by Type**: Displays the distribution of DHCP message types by count over a selected time range, using a pie chart segmented by the ***type.keyword*** field.

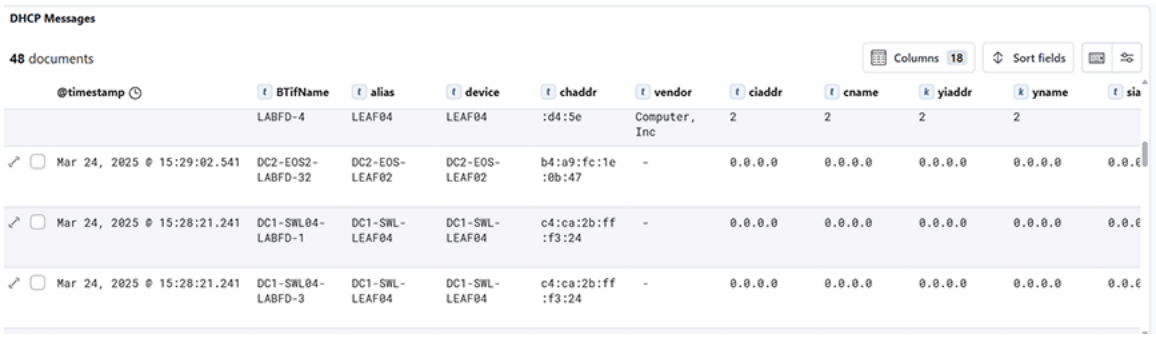
Figure 2-17: Production > DHCP Dashboard- Middle



The lower dashboard includes:

- **DHCP Messages**: Displays a time-series histogram and detailed document table of DHCP messages filtered by type over a selected time range.

Figure 2-18: Production > DHCP Dashboard- Bottom



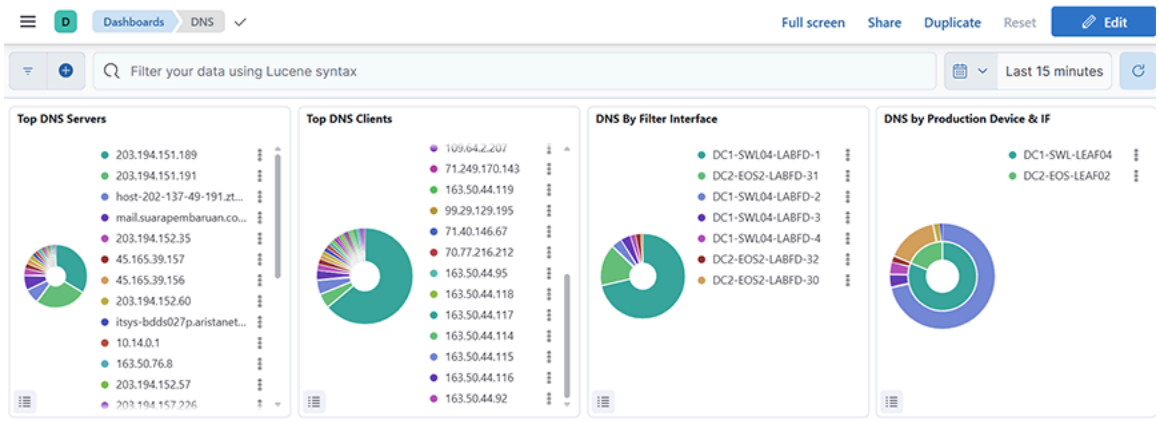
2.8 DNS

Selecting **Production** → **DNS** analyzes DNS activity on the **Analytic Node**.

The **DNS Dashboard** summarizes information from analyzing DNS activity and provides the following panels. The dashboard displays:

- **Top DNS Server**: Displays the top 25 DNS servers by request count over a selected time range, using a pie chart segmented by server name.
- **Top DNS Clients**: Displays the top 25 DNS clients based on query count over a selected time range, represented as pie chart segments grouped by client name.
- **DNS by Filter Interface**: Displays the top 60 DNS query sources based on the ***BTIfName*** field, counted and ordered by frequency over a selected time range.
- **DNS by Production Device & IF**: Displays DNS query counts grouped by production device and interface, using a nested split on device and devicePort fields over a selected time range.

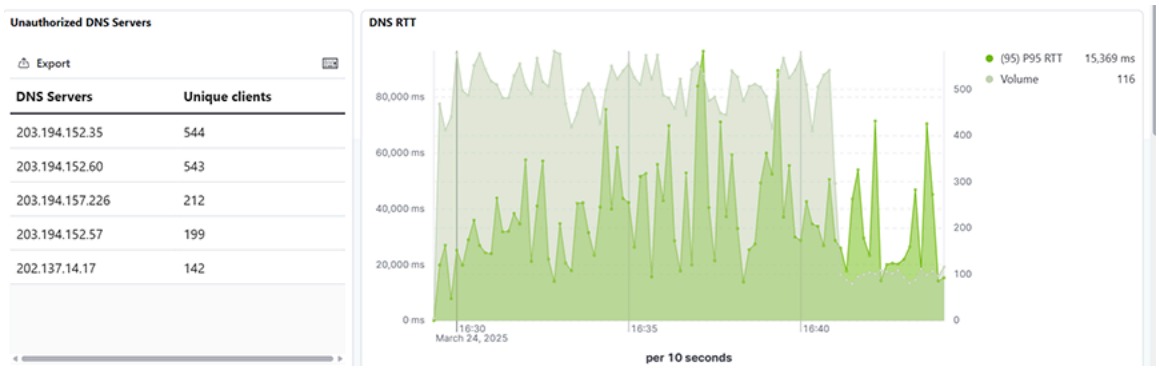
Figure 2-19: Production > DNS Dashboard - Top



The middle dashboard includes:

- **Unauthorized DNS Server**: Displays the top unauthorized DNS server, excluding a defined list of approved IP ranked by their unique client count over a selected time range.
- **DNS RTT**: Displays the 95th percentile of DNS Round-Trip Time (RTT) and the query volume, aggregated over time, for all sources over a selected time range.

Figure 2-20: Production > DNS Dashboard- Middle

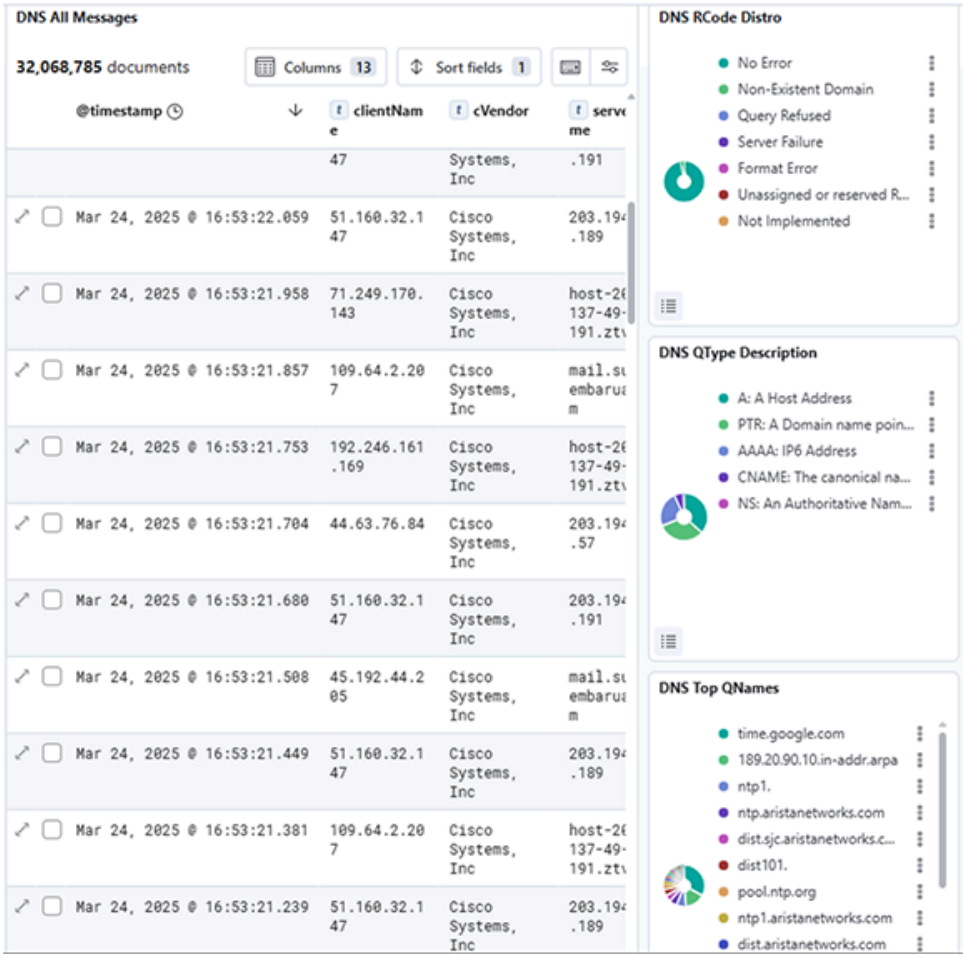


The lower dashboard includes:

- **DNS All Messages**: Displays all DNS-related log messages along with key fields like client IP, server name, RTT, and query types over a selected time range.
- **DNS Rcode Distro**: Shows the distribution of DNS Response Codes (RCode) based on their frequency over a selected time range.
- **DNS QType Description**: Shows the distribution of DNS Query Types (QTypes) based on their frequency over a selected time range.

- **DNS Top QNames**: Shows the top Queried domain Names (QNames) in DNS traffic, ranked by count over a selected time range.

Figure 2-21: Production > DNS Dashboard- Bottom



Note: The query and response packet timestamps compute the DNS RTT value. If a query packet does not answer by a response packet within ***180*** seconds, then the RTT value is set to ***-1***.

2.9 ICMP

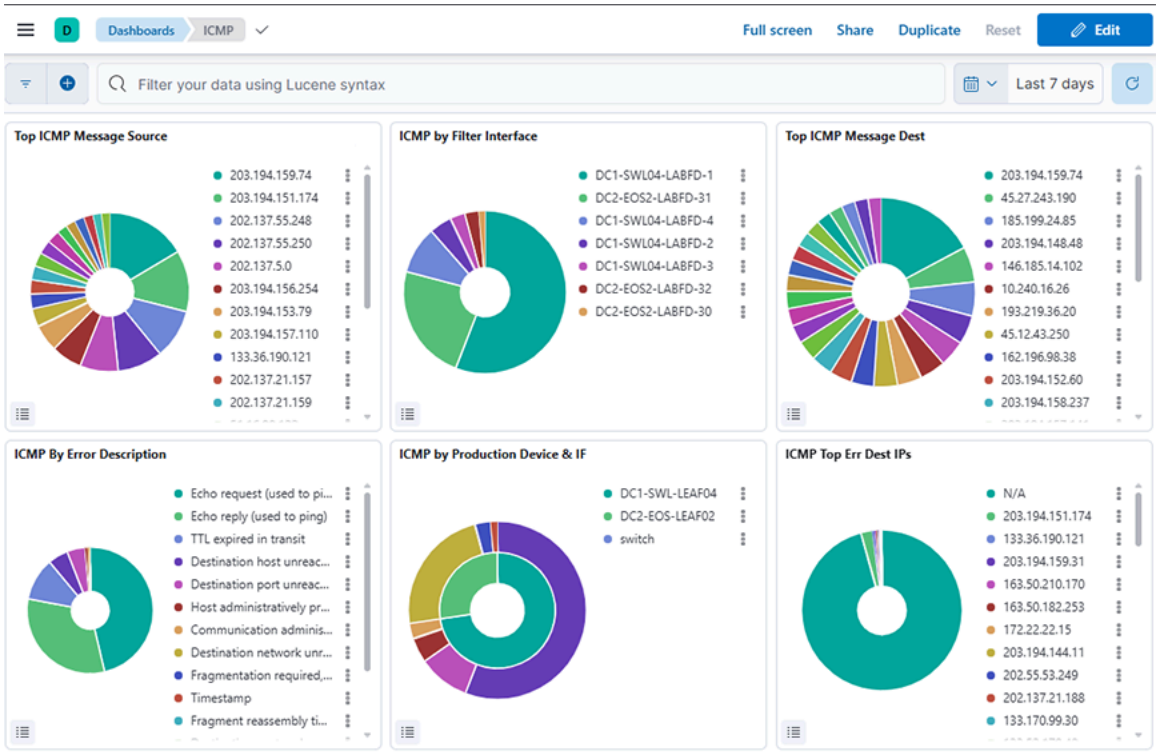
The **ICMP Dashboard** summarizes information from analyzing ICMP activity and provides the following panels.

The dashboard displays:

- **Top ICMP Message Source**: Displays the top source IPs sending ICMP messages, ranked by count over a selected time range.
- **ICMP by Filter Interface**: Displays the distribution of ICMP messages by filter interface (***BTIName***) over a selected time range.

- **Top ICM Message Dest**: Displays the top destination IP addresses for ICMP messages over a selected time range.
- **ICMP by Error Description**: Displays the distribution of ICMP messages categorized by error description over a selected time range.
- **ICMP by Production Device & IF**: Displays ICMP message distribution grouped by production device and interface over a selected time range.
- **ICMP Top Err Dest IPs**: Displays the top destination IPs associated with ICMP error messages over a selected time range.

Figure 2-22: Production > ICMP Dashboard - Top

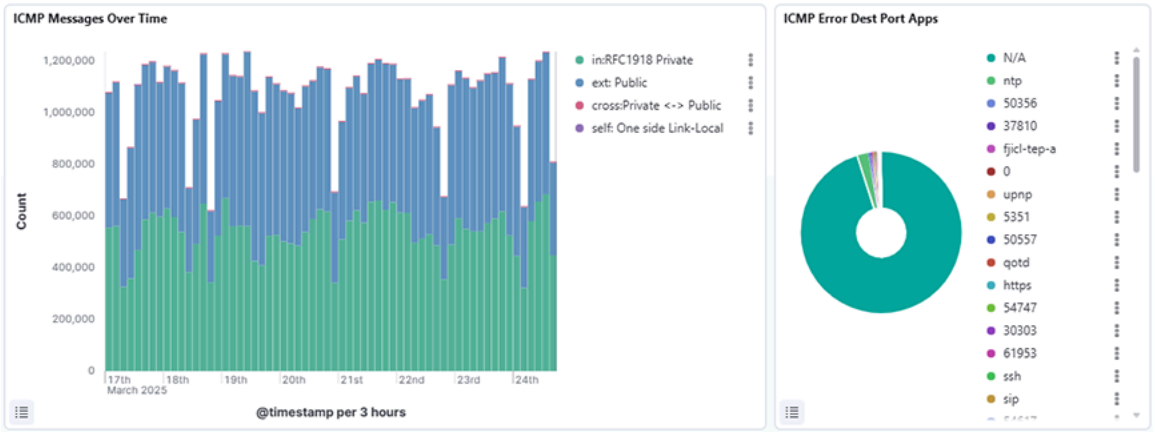


The middle dashboard includes:

- **ICMP Messages Over Time**: Displays the count of ICMP messages over a selected time range, broken down by traffic tags such as private, public, cross, and self, using a date histogram on the timestamp field.

- **ICMP Error Dest Port Apps**: Displays the distribution of destination port applications that triggered ICMP errors over a selected time range, using a pie chart grouped by the ***errPortApp.keyword*** field and sorted by descending count.

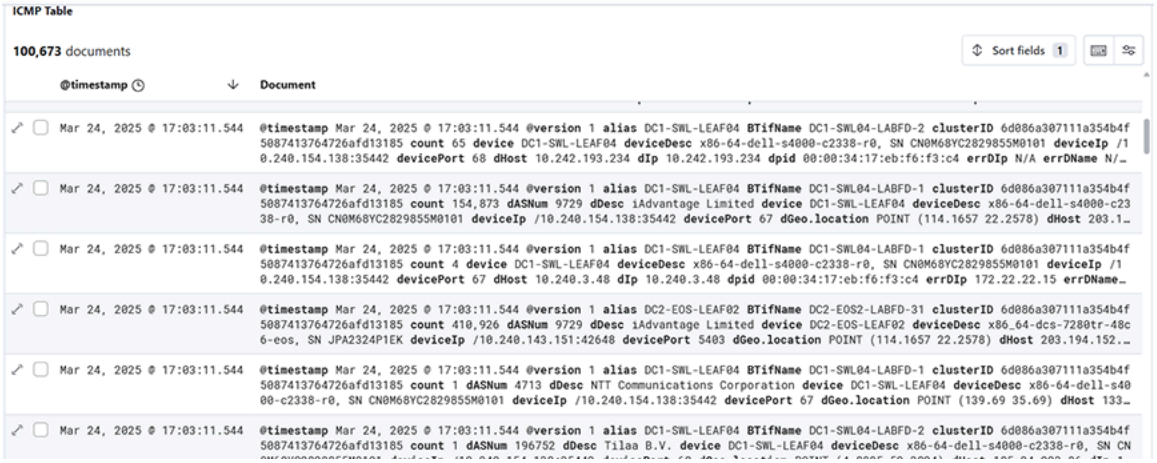
Figure 2-23: Production > ICMP Dashboard - Middle



The lower dashboard includes:

- **ICMP Table**: Displays a detailed table of ICMP packet metadata over a selected time range, including fields such as timestamp, source and destination IPs, device info, error descriptions, and message types for in-depth analysis.

Figure 2-24: Production > ICMP Dashboard- Bottom



Chapter 3

DMF

This chapter describes uses the dashboards on the **DMF** tab to monitor activity on the DANZ Monitoring Fabric Controller. It includes the following sections.

Note: Information displayed on these dashboards requires configuring an ACL for Redis and replicated Redis using the Analytics CLI after first boot configuration.

3.1 Policy

The **DMF** → **Policy** dashboard summarizes information about DANZ Monitoring Fabric policy activity and provides the following panels:

- **Top Active Policies**: Displays the top active policies based on the sum of bit rates over a selected time range, helping identify which policies contribute the most to traffic.
- **Top Filter Interfaces**: Displays the top filter interfaces based on the sum of bit rates over a selected time range, allowing identification of interfaces handling the most traffic.
- **Top Service Interfaces**: Displays the top service interfaces based on the sum of bit rates over a selected time range, highlighting which service interfaces handle the most traffic.
- **Top Delivery Interfaces**: Displays the top delivery interfaces based on the sum of bit rates over a selected time range, helping to identify which interfaces are handling the most traffic for policy delivery.

- **Mean Bit Rate**: Shows the mean bit rate over a selected time range by summing bit rates across different policy types (filter, delivery, and service) and plotting the values against time.

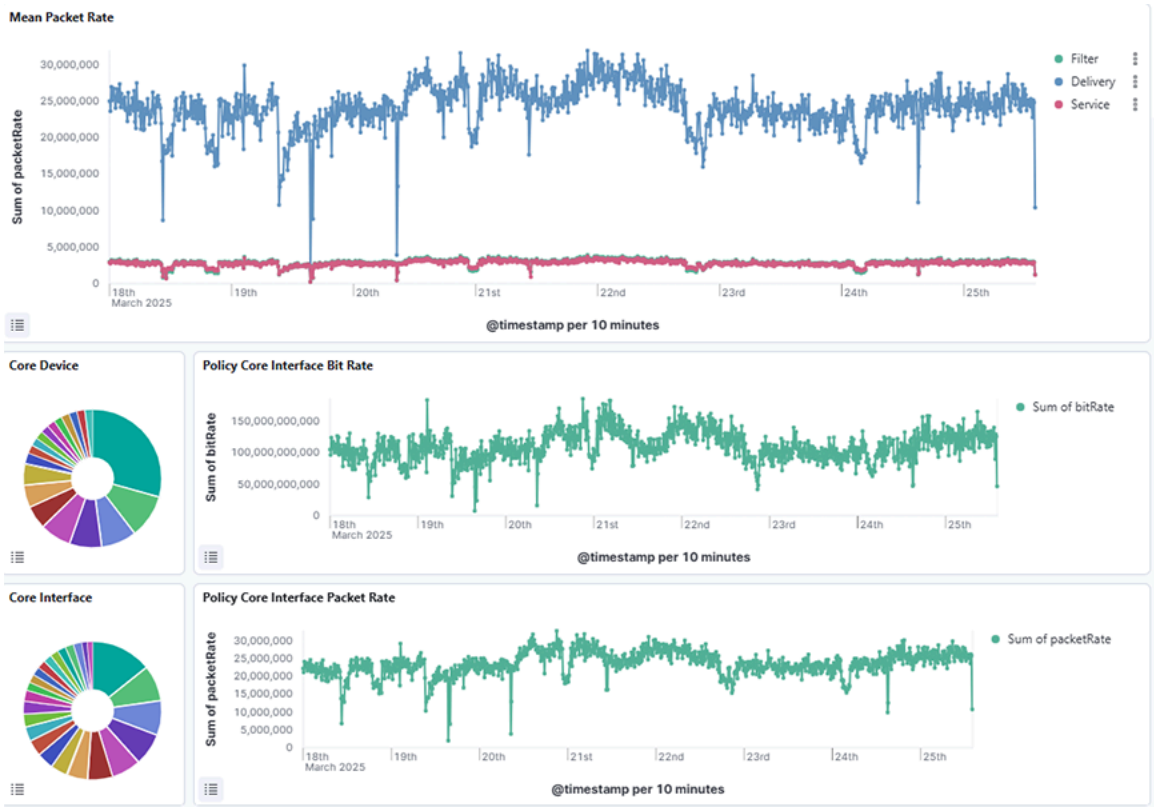
Figure 3-1: DMF > Policy Dashboard - Top

The middle dashboard includes:

- **Mean Packet Rate**: Shows the mean packet rate over a selected time range by summing packet rates across different policy types (filter, delivery, and service) and plotting the values against time.
- **Core Device**: Shows the distribution of bit rate across core devices over a selected time range, with data aggregated by device alias names and ordered by the total sum of bit rate.
- **Policy Core Interface Bit Rate**: Shows the sum of bit rates over time for core interfaces related to policy traffic, aggregated by timestamp over a selected time range.
- **Core Interface**: Shows the distribution of bit rate across core interfaces over a selected time range, with data aggregated by interface names and ordered by the total sum of bit rate.

- **Policy Core Interface Packet Rate**: Shows the sum of packet rates over time for core interfaces associated with policy traffic, aggregated by timestamp over a selected time range.

Figure 3-2: DMF > Policy Dashboard - Middle

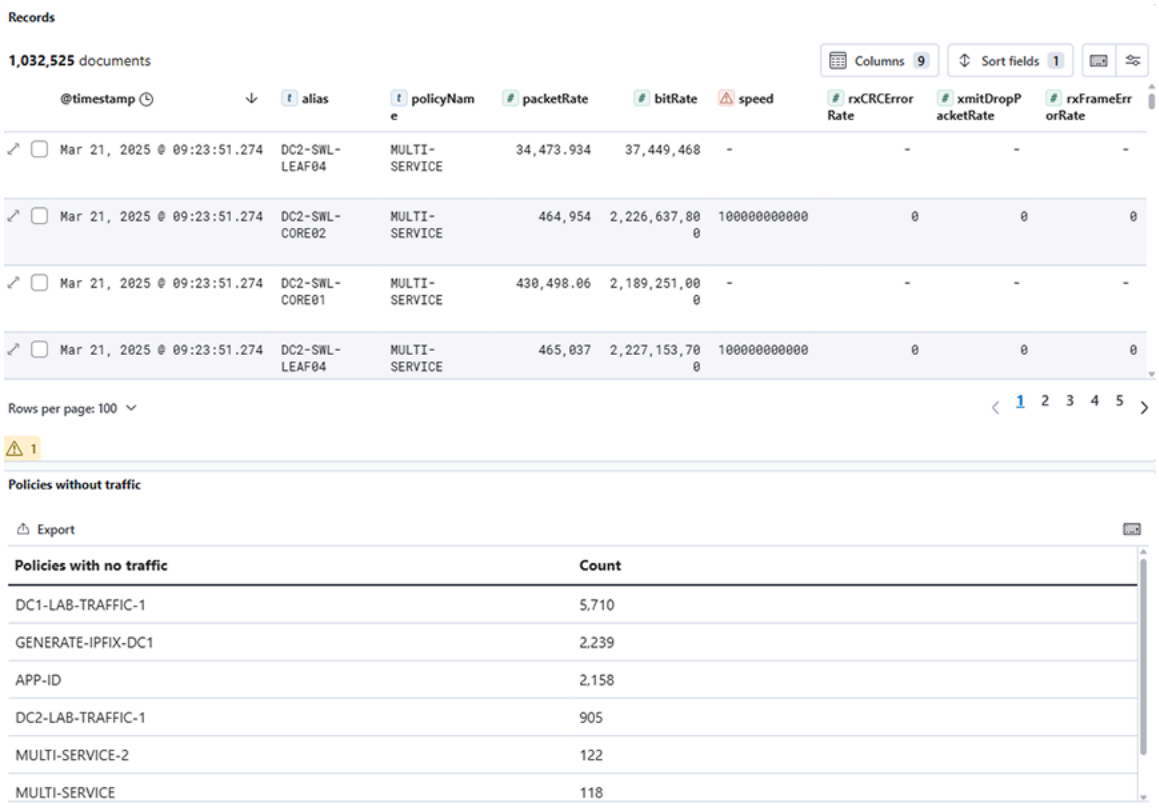


The lower dashboard includes:

- **Records**: Displays detailed raw records of policy-related statistics across various types over a selected time range, including metadata fields such as timestamps, types, bit rates, and packet counts.

- **Policies with no traffic**: Lists policies with zero traffic by counting occurrences of policy names where the bit rate is zero, aggregated over a selected time range.

Figure 3-3: DMF > Policy Dashboard - Bottom



Use the **Top Active Policies** visualization to verify that your DANZ Monitoring Fabric policies are active and behaving as expected.

Use the **Filter Interfaces** visualization to balance the utilization of your filter interfaces and ensure that it doesn't drop any packets to analyze.

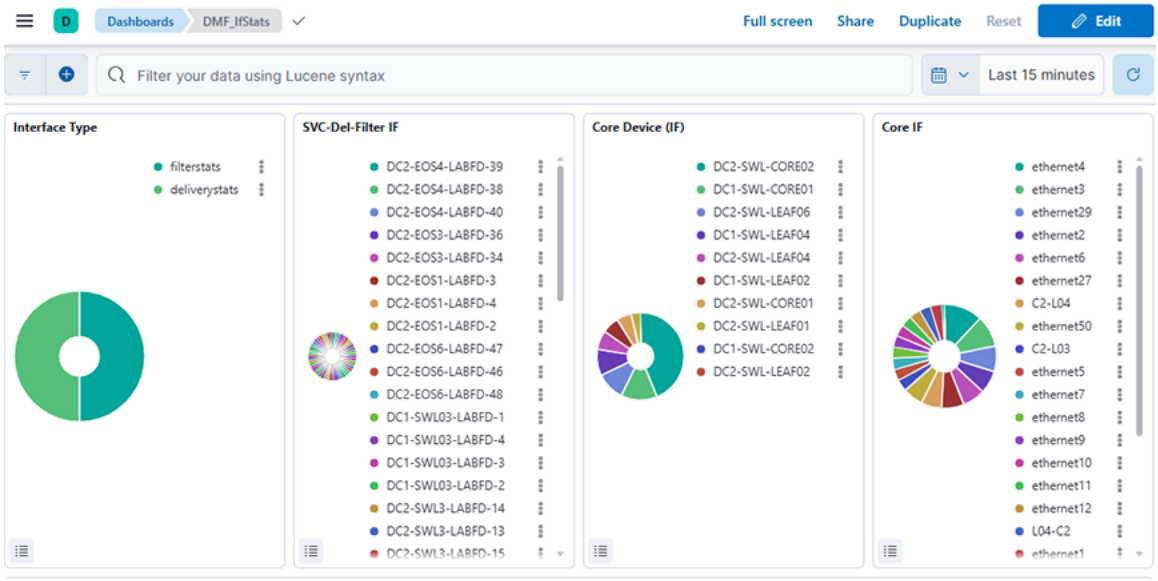
3.2 Interface

Click the **DMF** → **Interface** tab to display the following dashboard.

- **Interface Type**: Displays the distribution of interface types by summing the bit rate for each type over a selected time range.
- **SVC-Del-Filter IF**: Displays the distribution of traffic across different service, delivery, and filter interfaces by summing the bit rate for each interface over a selected time range.
- **Core Service (IF)**: Displays the sum of transmitted bit rates (txBitRate) for different core device interfaces over a selected time range.

- **Core IF**: Displays the sum of transmitted bit rates (txBitRate) across core interfaces, grouped by interface name, over a selected time range.

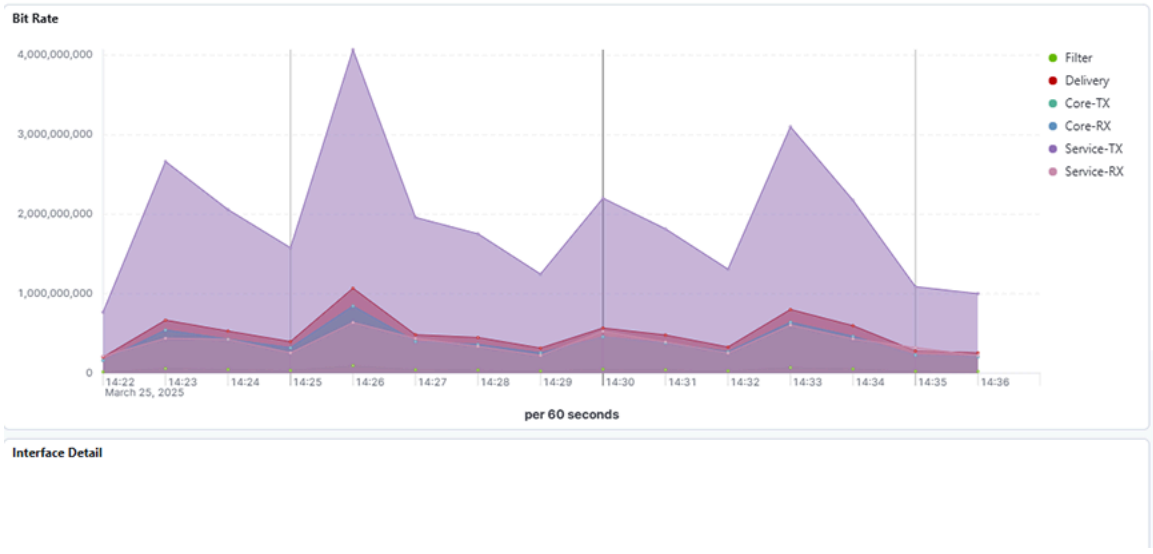
Figure 3-4: DMF > Interface Dashboard - Top



The lower dashboard includes:

- **Bit Rate**: Displays the average bit rates for filter, delivery, core transmit (TX), core receive (RX), service transmit (TX), and service receive (RX) interfaces over a selected time range.
- **Interface Detail**: Displays detailed interface metrics, including bit rate, packet count, error rates, and other network statistics over a selected time range.

Figure 3-5: DMF > Interface Dashboard - Bottom

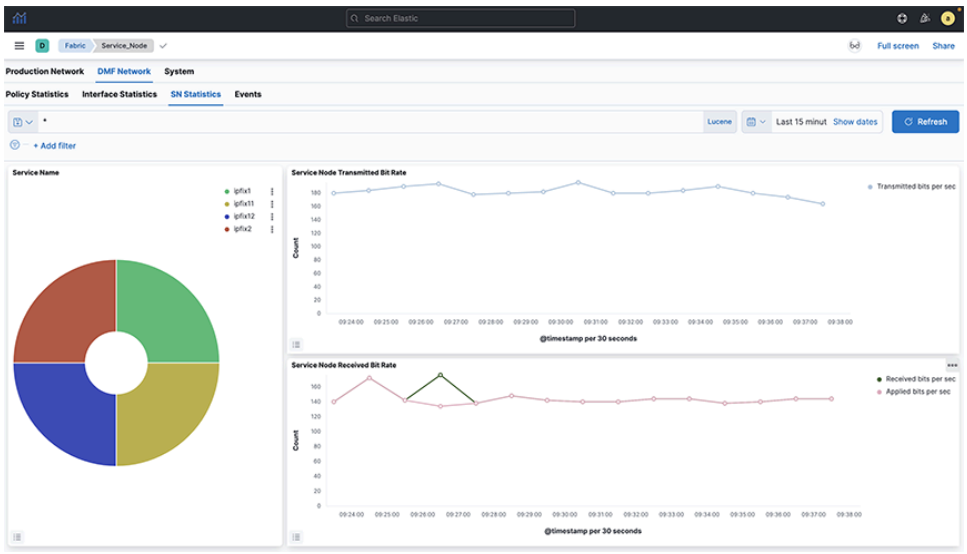


3.3 SN (Service Node)

Click the **DMF** → **SN** tab to display the following panels:

- **Service Name**: Displays the count of records grouped by service name over a selected time range.
- **Service Node Transmitted Bit Rate**: Displays service nodes' average transmitted bit rate per second over a selected time range.
- **Service Node Received Bit Rate**: Displays the average received and applied bit rates per second for service nodes over a selected time range.

Figure 3-6: DMF > SN Dashboard



Select a service from the pie chart to see the statistics for a specific managed service. It will display statistics for the selected service.

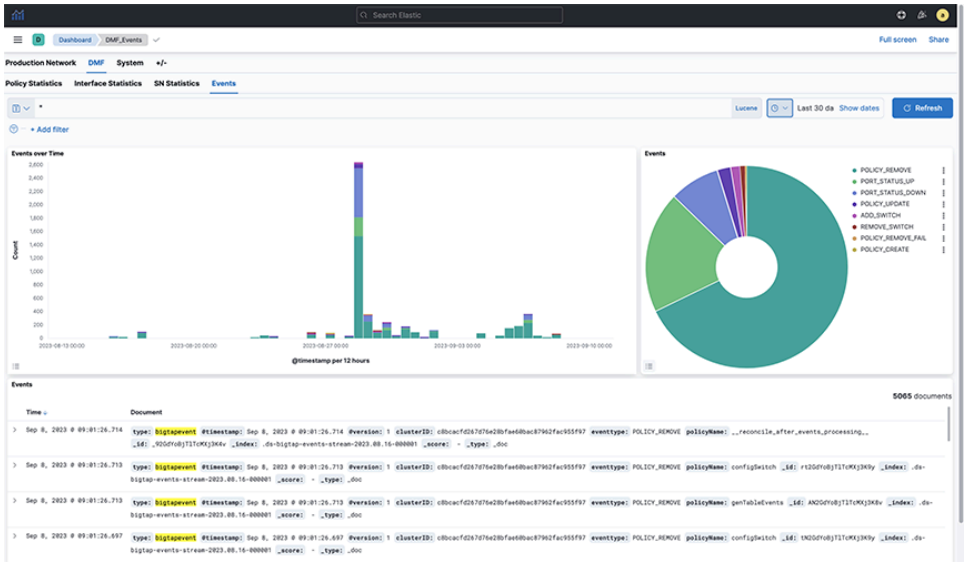
3.4 Events

The **DMF** → **Events** dashboard summarizes information about DANZ Monitoring Fabric management network events and provides the following panels:

- **Events Over Time**: Displays the count of different event types over a selected time range using a date histogram on the timestamp field.
- **Events**: Displays the count of different event types over a selected time range, based on the ***eventtype.keyword*** field.

- **Events**: Displays a timeline of ***bigTapevent*** records, showing the distribution of different event types and their details over a selected time range.

Figure 3-7: DMF > Events Dashboard



Chapter 4

System

This chapter describes the dashboards on the **System** tab to configure the DANZ Monitoring Fabric Controller and alerts. It includes the following sections.

- Configuration
- Status
- About

4.1 Configuration

This dashboards on the **Configuration** tab configures the alerts and analytics for the DANZ Monitoring Fabric Controller for different settings and configurations. It has the following sections:

- Configure Alerts
- Analytics Configuration

4.1.1 Configure Alerts

This dashboards on the **Configuration** tab configures the alerts for the DANZ Monitoring Fabric Controller for different settings. It has the following settings:

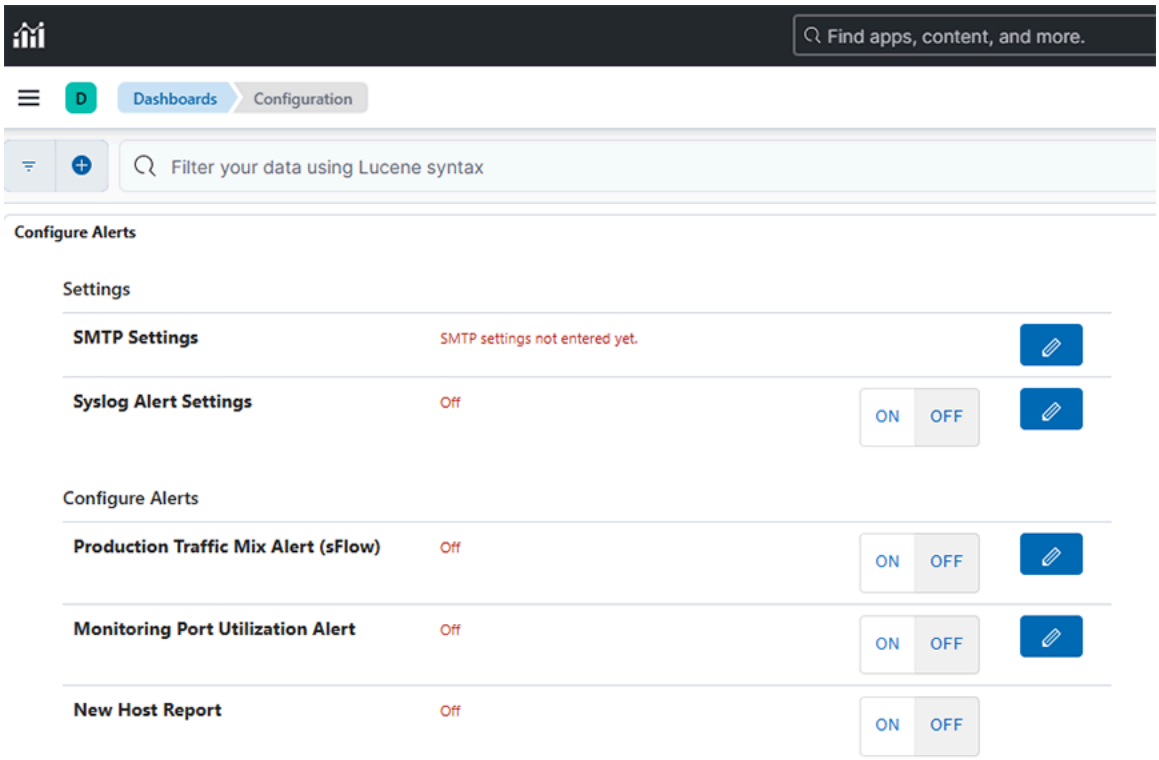
- SMTP Settings
- Syslog Alert Settings

The **Configure Alert** tab has following options:

- Production Traffic Mix Alert (sFlow)
- Monitoring Port Utilization Alert

- New Host Report

Figure 4-1: System > Configure Alerts



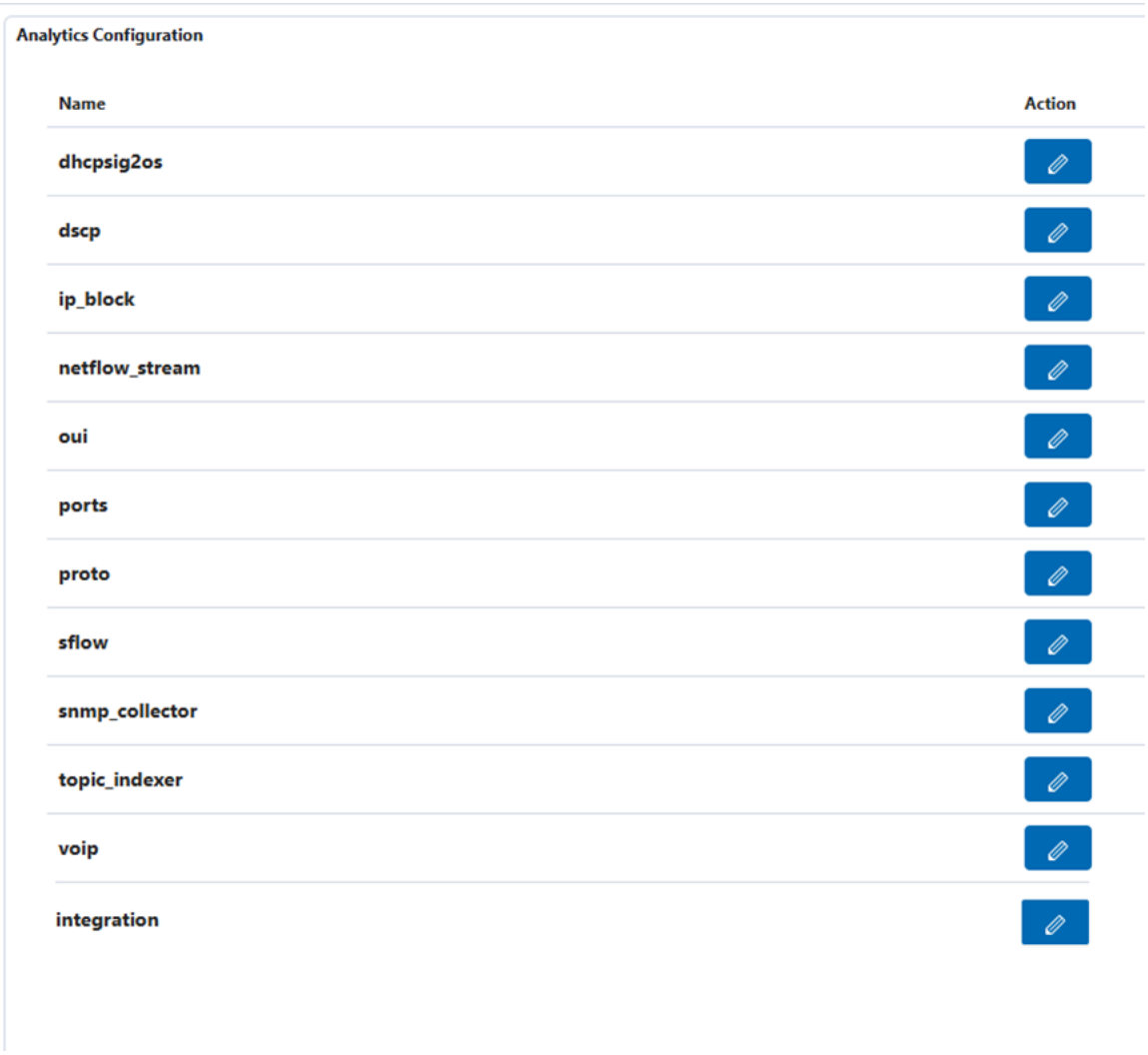
4.1.2 Analytics Configuration

This tab describes the configurations of Arista Analytics. It includes the following sections:

- DHCP to OS
- IP Address Blocks
- NetFlow Stream
- OUI
- Ports
- Protocols
- SNMP Collector
- Topic Indexer

- Integration

Figure 4-2: System > Analytic Configuration



4.1.2.1 DHCP to OS

DHCP signatures can map to known operating systems. These unique signatures are from https://www.fingerbank.org/. As shown in the following image, several two-digit numbers are assumed signatures of each OS (derived from ***fngerbank.org***).

Figure 4-3: Unique OS Signatures from fingerbank.org

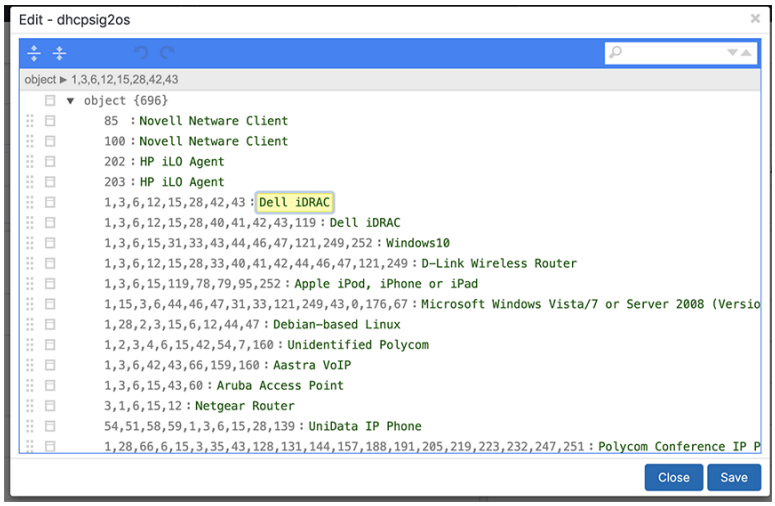
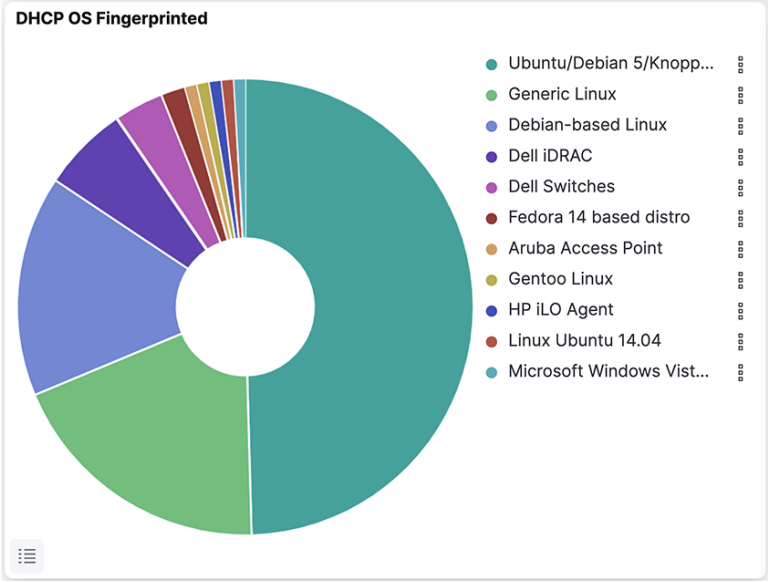


Figure 4-4: OS Information Received through DHCP Signatures



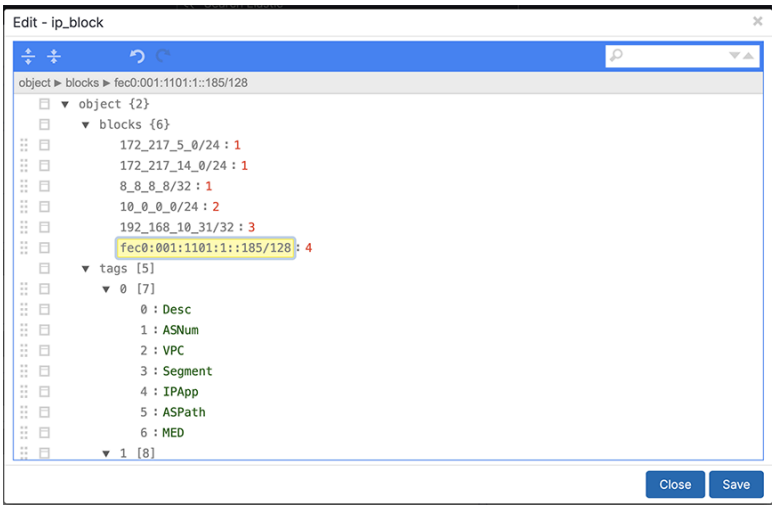
4.1.2.2 IP Address Blocks

Map an IP address or a range of addresses to a description, which searches for description text instead of the IP address. This feature identifies a specific group or organization sending or receiving traffic.

Complete the following steps to assign a single IP address or a block of IP addresses to a tool, group, or organization.

1. Select **System** > **Configuration** and click the **Edit** control to the left of the IP Block section.

   Figure 4-5: Edit IP Blocks

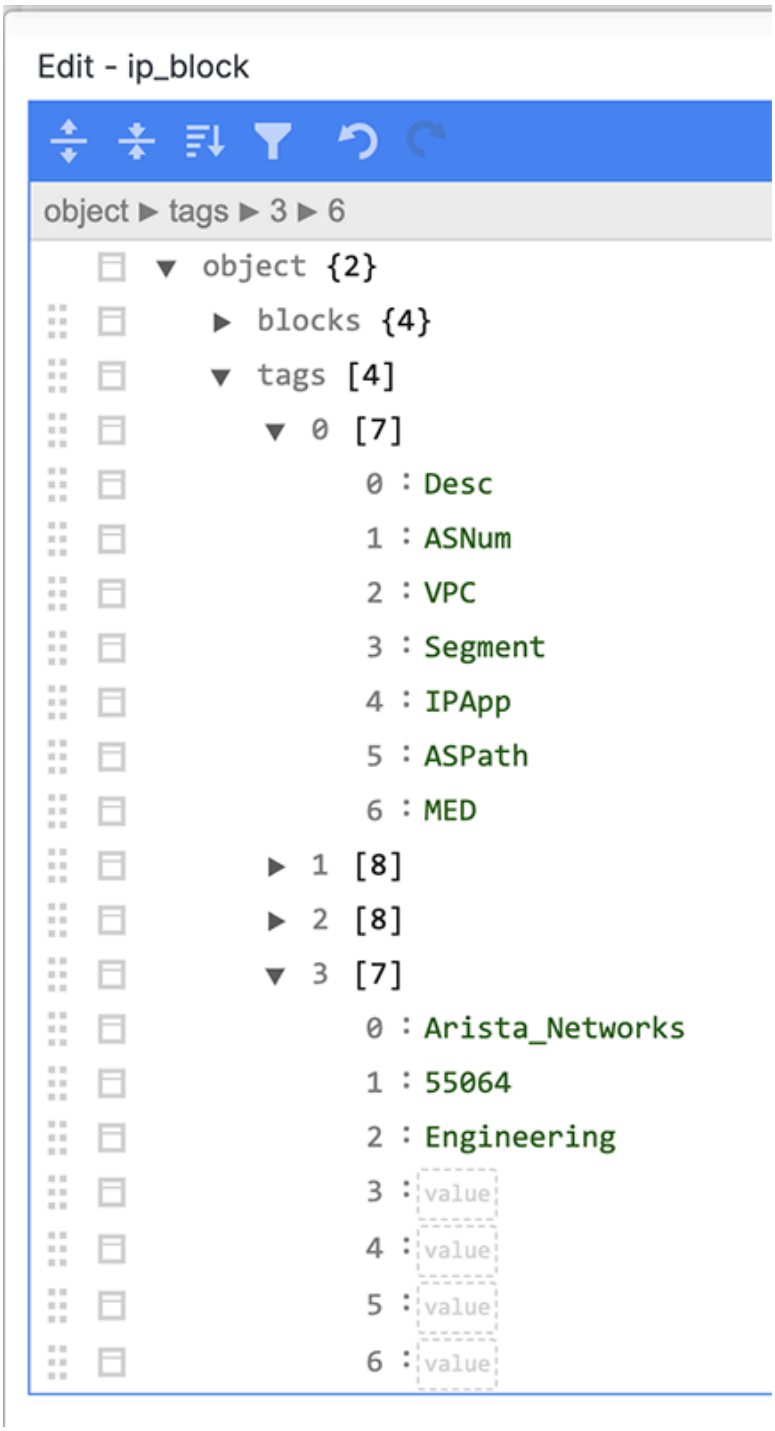


2. Copy an existing block by clicking on any square box along the left and select **Duplicate** from the pop-up menu.

   The duplicated block will be appended to the existing block list and assigned the next numerical sequence identifier.

3. Scroll down to the end of the tags section to the numerical identifier assigned to the new block.

Figure 4-6: Key Value Pairs



It automatically copies the first four keys. The purpose of each of these default keys is as follows.

- **Desc**: A short descriptive text entry.
- **ASNum**: Automatically populated with the BGP Autonomous Systems (AS) numbers for well-known networks.
- **VPC**: Virtual Private Cloud (tenant), automatically populated with the VPCs used in an integrated Converged Cloud Fabric network.

- **Segment**: Network segment within a Converged Cloud Fabric VPC.

To identify a user, application, tool, group, or organization, use the ***Desc*** key. You can leave the other fields blank.

4. Type a value for the Desc key in double quotation marks (“).
5. (Optional) To define an additional key, select any key and choose **Duplicate** from the pop-up menu. Then, type over the existing value with the correct value for the new key.

   Existing dashboards use the default keys. The customized dashboards can use added key pairs. The fifth and sixth keys can be custom.

   The **source** and **destination** IPv4 address have these keys for the flow. For example, the source description would be ***sDesc***, and the destination description would be ***dDesc***.

   Note: Remember to match values in the same order as the corresponding key positions.

4.1.2.3 NetFlow Stream

Arista Analytics may consolidate NetFlow records to improve performance.

The Analytics server/cluster consolidates flows received within two seconds into a single flow when the source and destination IP addresses are the same, or the source or destination L4 protocol port is the same.

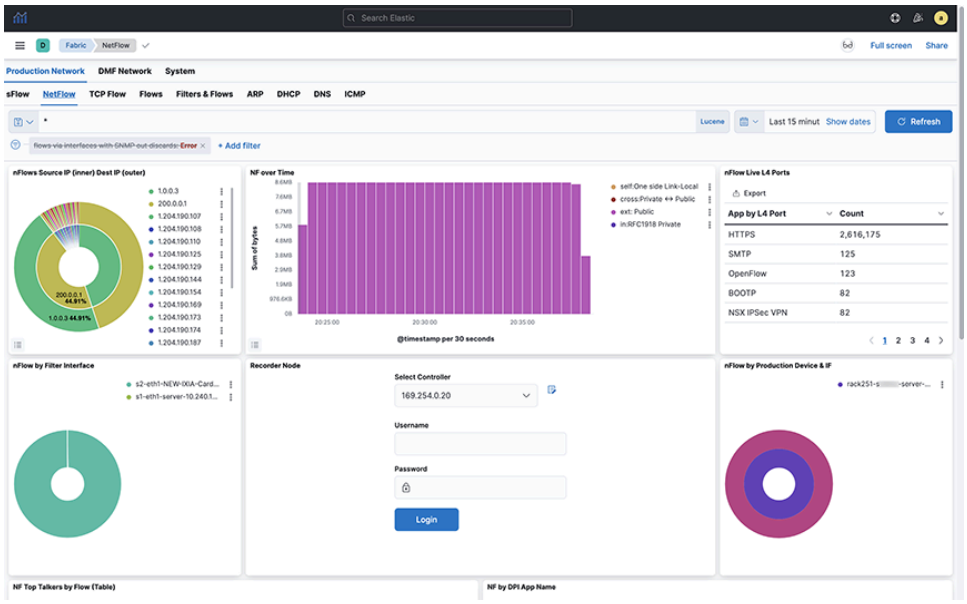
For example, ten flows received by the Analytics server within 30 seconds are consolidated into a single flow if the source and destination IP addresses and destination port are the same for all the flows and only the source ports are different or if the source and destination IP addresses and source port are the same for all the flows and only the destination ports are different. This consolidated flow displays as a single row.

By default, the NetFlow Optimization is enabled for Netflow v5 and disabled for Netflow v9 and IPFIX. To allow the Netflow Optimization for Netflow v9 and IPFIX, refer to the Netflow v9/IPFIX Records section.

This consolidation improves Analytics NetFlow performance, allowing more efficient indexing and searching of NetFlow information.

The following figure shows the **NF Detail** window on the **NetFlow** dashboard, which provides an example of NetFlow information with optimization.

Figure 4-7: Analytics NetFlow Optimization



4.1.2.3.1 NetFlow Traffic from Third-party Devices

This section displays third-party device and interface names. It shows hop-by-hop forwarding of flows when NetFlow traffic comes from a third-party device. For a query for a specific flow, Arista Analytics shows the device and interface names associated with that flow. If the flows go through two hops, it displays the device and interface names associated with flows.

Arista Analytics can act as a NetFlow collector for third-party devices. In this case, Arista Analytics displays third-party device management IP addresses and the interface index (iFindex) of the interface for each NetFlow-enabled third-party device.

For example, the **nFlow by Production Device & IF** window shows that ***10.8.39.198*** is the third-party device that forwards NetFlow traffic. The iFindex of the interface on that device where NetFlow is enabled is ***0***, ***2***, ***3***, ***4***.

To discover the device name and the actual interface name rather than the iFindex, Arista Analytics automatically does an SNMP walk by getting the third-party device management IP from flow information. By default, Analytics uses the SNMP community name **public** to get the device name and interface name. If the SNMP community name of the third-party device is not **public**, change it in the Arista Analytics SNMP collector configuration.

Note: ***Analytic Node DMF 8.3.0*** release supports both SNMPv2 and SNMPv3.

Note: For IPFIX and nFlow v9, configure the third-party device to send the iFindex. The Analytics node will do an SNMP walk to get the interface names associated with that iFindex. By default, it does not send the iFindex with IPFIX or nFlow v9. For example, to send the iFindex for IPFIX and nFlow v9, enable **`match interface input snmp`** and **`match interface output snmp`** under **flow record** configuration on the third-party device.

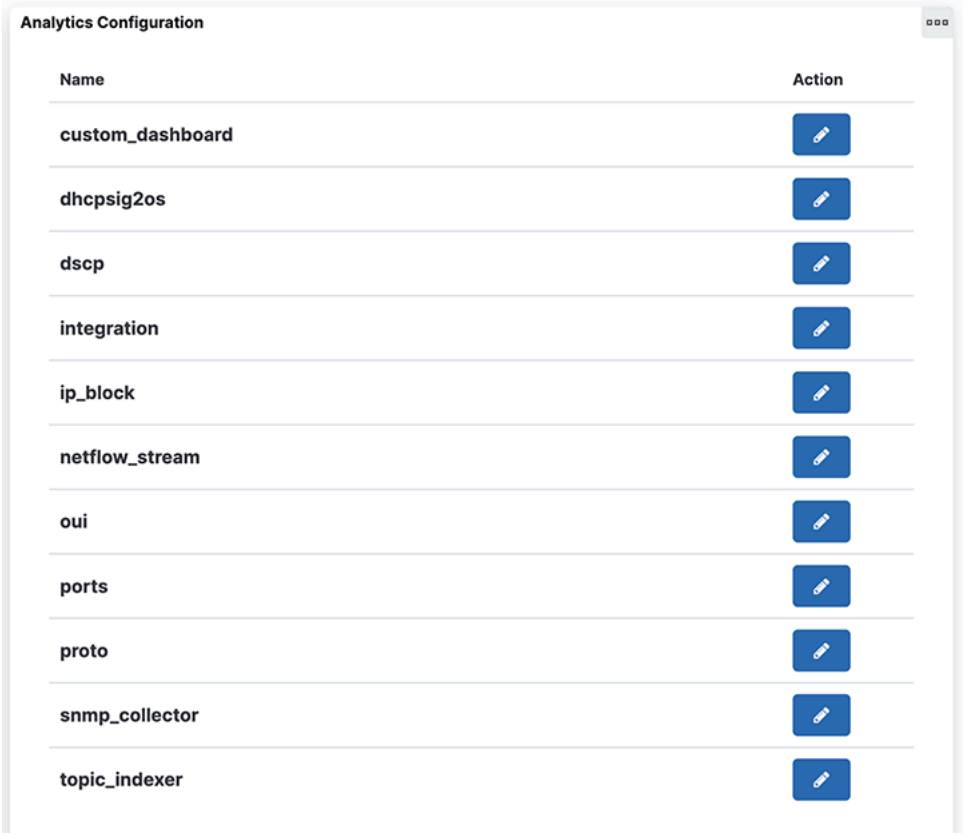
DMF Analytic > **System** > **Configuration** > **Analytic Configuration** > **snmp_collector**

Arista Analytics then performs SNMP polling and displays the third-party device name and the actual interface name in the **nFlow by Production Device & IF** window.

To perform the SNMP configuration, complete the following steps:

1. On the screen shown later, click **DMF Analytic** > **System** > **Configuration** > **Analytic Configuration** > **snmp_collector** > **Edit**.

   Figure 4-8: Analytic snmp_collector config



The system displays the following edit dialog.

Figure 4-9: Analytic Configuration > snmp_collector > Edit Dialog (SNMPv2 Configuration)

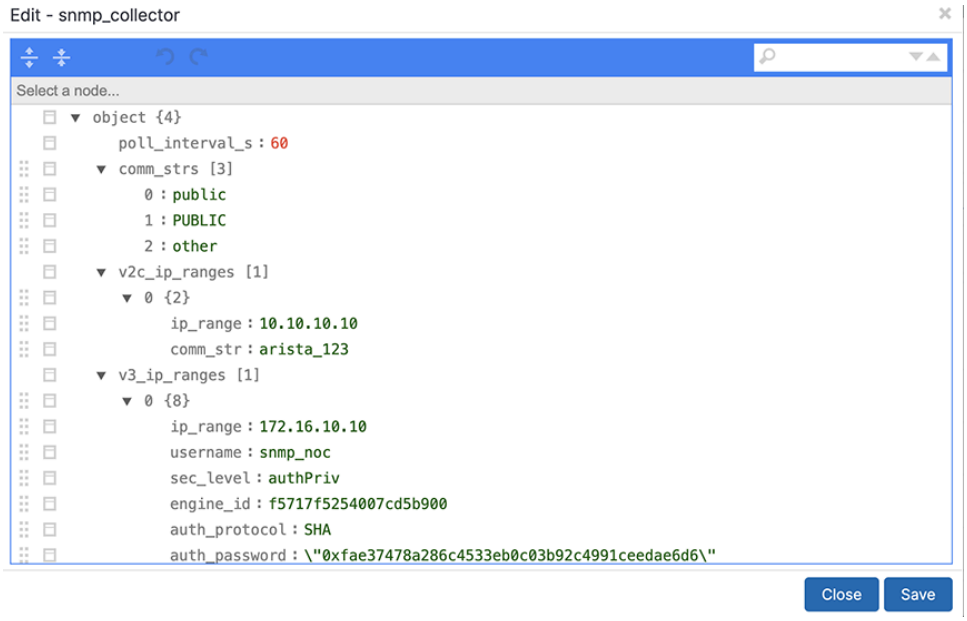
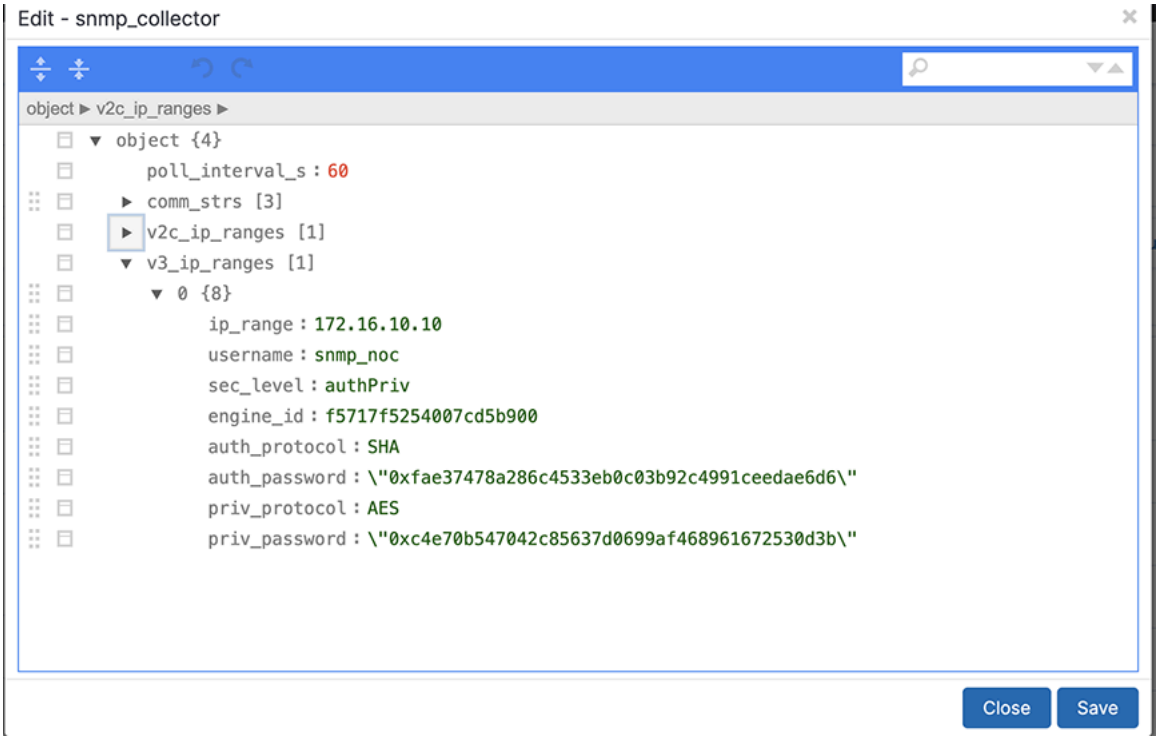


Figure 4-10: Analytic Configuration > snmp_collector > Edit Dialog (SNMPv3 Configuration)

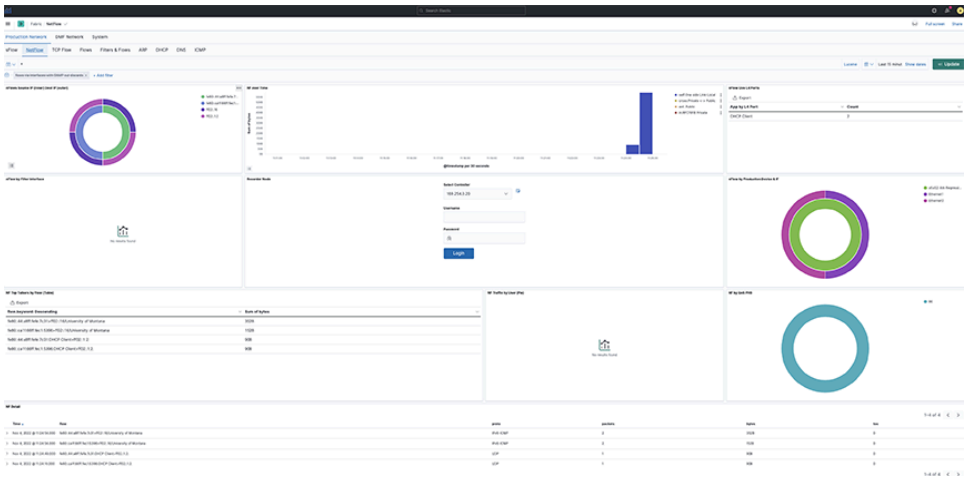


2. Click the community string **public** to change it to a different value, as shown in the following dialog.

   By default, the SNMP collector polls devices every ***60*** seconds.
3. To change the SNMP poll interval, click the value ***60***, change it to the preferred value, and click **Save**.

After completing this configuration, the third-party device is polled for the device name and interface name, **nFlow by Production Device & IF** window displays it.

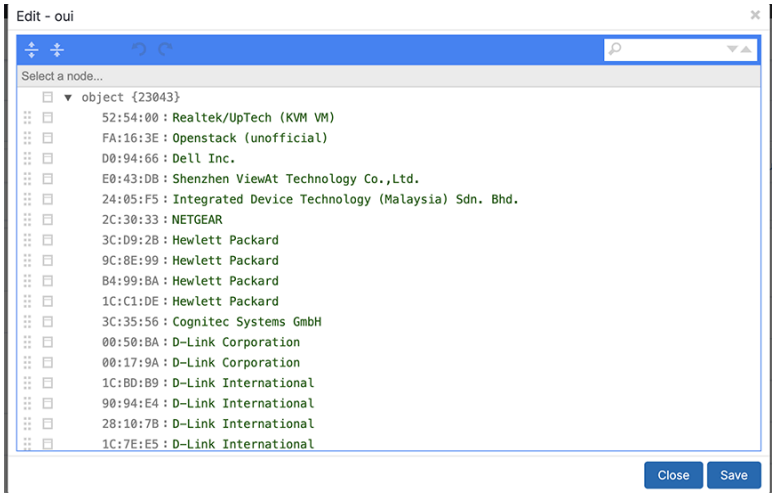
Figure 4-11: Analytic Configuration > snmp_collector > Edit Dialog



4.1.2.4 OUI

Map ARP Organizational Unique Identifiers (OUIs) for various hardware vendors.

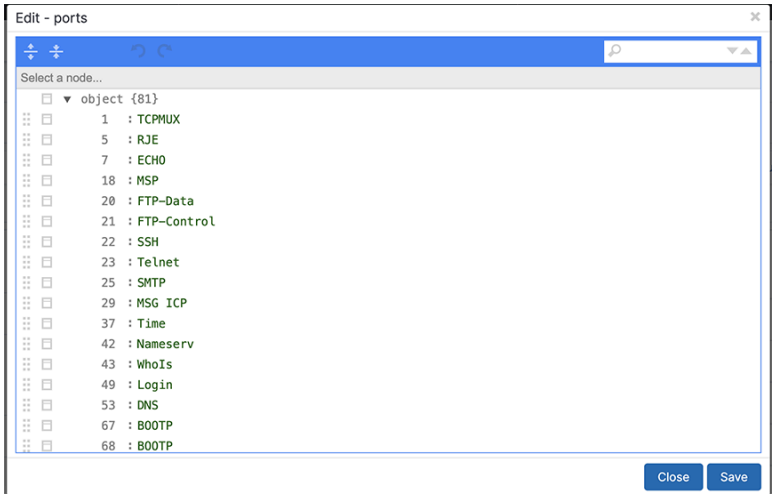
Figure 4-12: OUIs of Various Hardware Vendors



4.1.2.5 Ports

The Analytics Node maps typically used ports for their L4 applications.

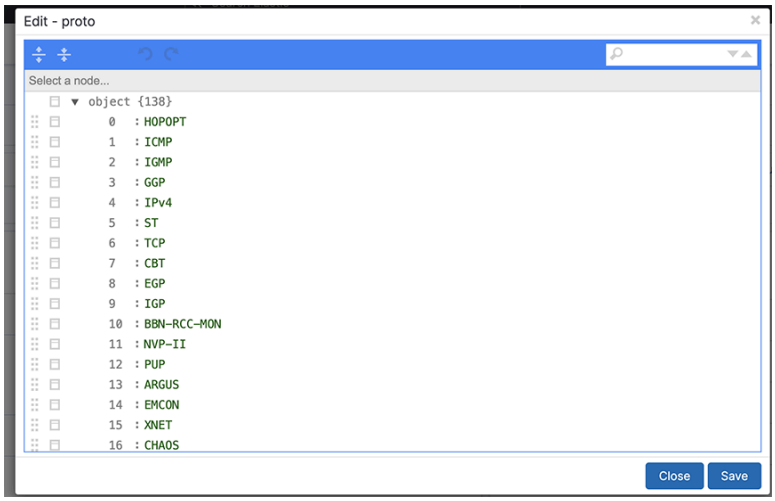
Figure 4-13: Edit Ports



4.1.2.6 Protocols

The Analytics Node maps protocols as following. These protocols can also be user defined for custom application custom protocols.

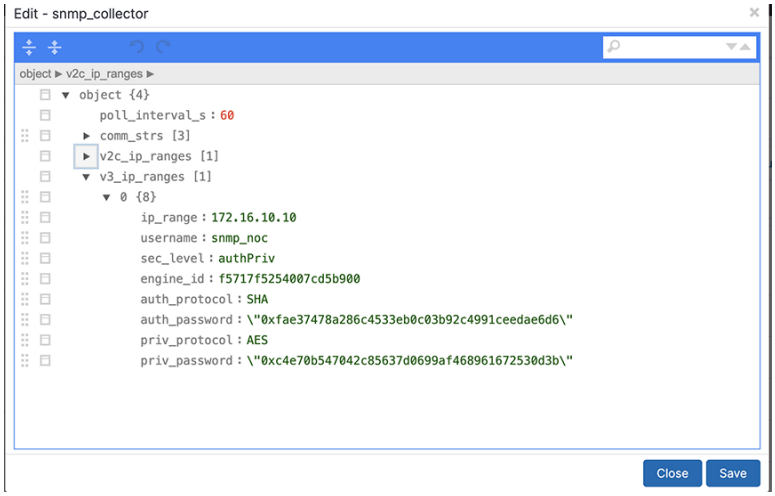
Figure 4-14: Edit Protocols



4.1.2.7 SNMP Collector

SNMP collectors facilitate third-party NetFlow or IPFIX sources. The Analytics Node supports both SNMPv2 and SNMPv3.

Figure 4-15: SNMP Collector



4.1.2.8 Topic Indexer

Description

The Analytics Node (AN) incorporates a feature known as topic_indexer, designed to facilitate data ingestion from customer Kafka topics and its subsequent storage into Elasticsearch indices.

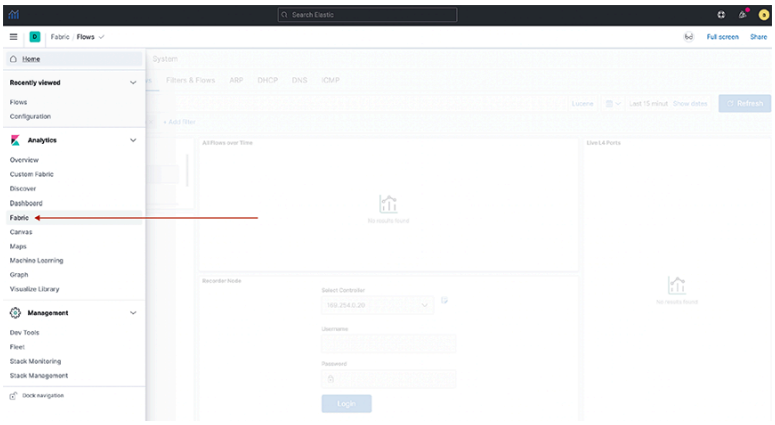
This process involves modifying field names and specifying the supported timestamp field during the ingestion phase. The renaming of field names enables the creation of dashboards used to visualize data across multiple streams, including DNS and Netflow.

The resulting indices can then be leveraged as searchable indices within the Kibana user interface, providing customers with enhanced search capabilities.

Implementation Details

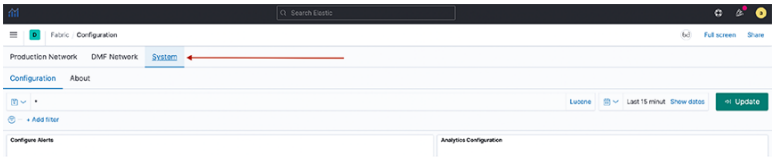
- Configure a stream job using topic_indexer. Access the setting via the Kibana dashboard in the analytics node.

- Locate the **topic_indexer** configuration on the Fabric Dashboard: **Analytics** > **Fabric** > **System** > **Analytics Configuration**, as shown in the following screenshots.
- **Figure 4-16: Analytics > Fabric**

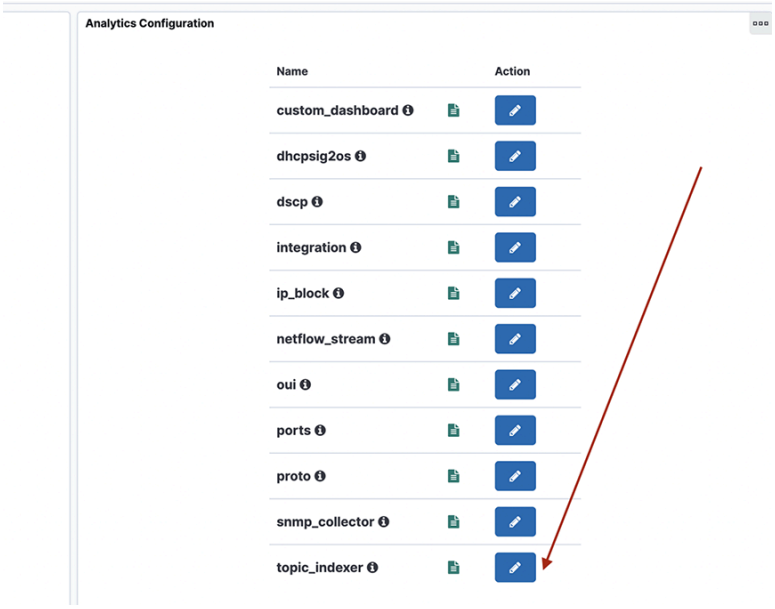


- Another view:

Figure 4-17: System > Analytics Configuration



- The design section shows the configuration for a topic.
- **Figure 4-18: Node selection**

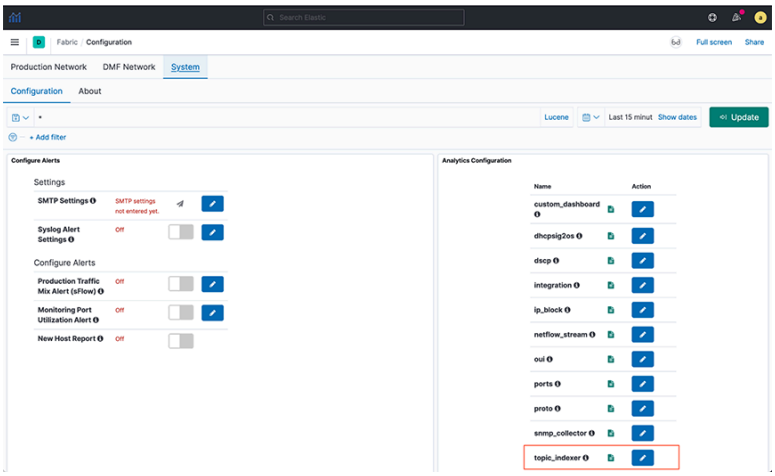


Configuration

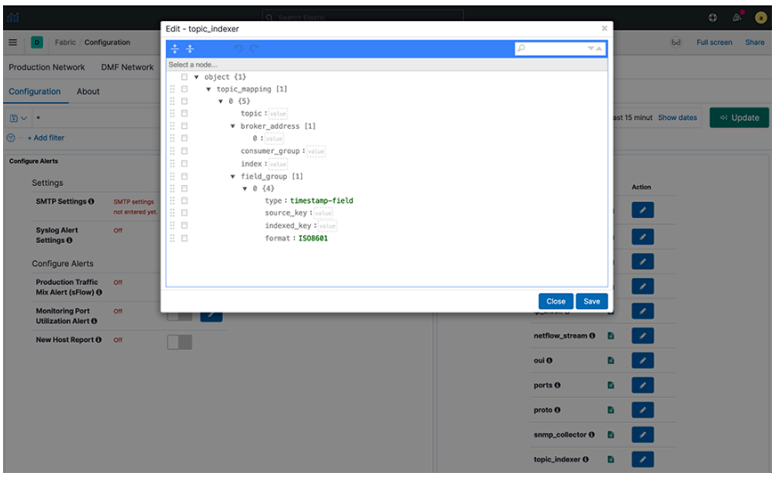
Kibana Configuration

- To perform the **topic_indexer** configuration, select the **System** > **Configuration** > **Fabric** page and open the **Analytics Configuration** panel:

  Figure 4-19: System > Configuration



- **Figure 4-20: Topic_indexer configuration**



Field Details

Each topic maps in JSON with the following fields:

- **topic**: Kafka topic name; type string and is a mandatory field.
- **broker_address**: Broker address(es), this is of type array; this will be of format [***IPv4|hostname:Port number***] and is a mandatory field.
- **consumer_group**: This is an optional field; however, there is always a consumer group if not specified explicitly in the configuration. It is **topic_name + index_name**. Setting this field is particularly useful when ingesting multi-partitioned topics from the client's end.
- **index**: A dedicated index name for the topic; type string. In Elastic Search (ES), it is created as ***topic_indexer_<index_name>*** and is a mandatory field.
- **field_group**: An optional JSON field mapping to specify any column rename/format transformations. It specifies the format for modifications to incoming data.
- **type**: To set the timestamp field as the type.
- **source_key**: The source field name in the incoming data.

- **indexed_key**: The destination field name inserted in the outgoing ES index.

  The **indexed_key** may be a @timestamp field of an ES index. If you do not specify a @timestamp field, ***topic_indexer*** automatically picks the received time of the message as the @timestamp of that message.
- **format**: Data format for the field (ISO8601).

Standards and Requirements

Input fields naming convention:

- Kafka allows all ASCII Alphanumeric characters, periods, underscores, and hyphens to name the topic. In***topic_indexer***, legal characters include: ***a-z0-9\\._\\-***
- Note that the only restriction ***topic_indexer*** has is on capitalizing topic names. ***topic_indexer*** does not support case-sensitive names. By default, ***topic_indexer*** treats the name as a lowercase topic. Hence, topic names should be lowercase only.
- All numeric names are also invalid field text.

Note: These conventions are valid for all other input types as well.

Examples of names:

Valid text:

- my-topic-name
- my_topic_name
- itlabs.mytopic.name
- topic123
- 123topic
- my-index-name

Invalid text:

- myTopicName
- ITLabs-Website-Tracker
- 12435
- MY-Index-name

Broker Address Format:

- A broker address in Kafka comprises two values: IPv4 address and Port Number.
- When entering the broker address, select the format ***IPv4:PORT.***

Application Scenario

Querying Across DataStream using runtime-fields

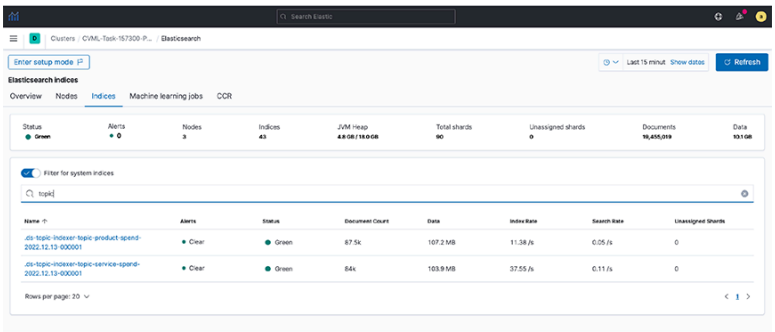
Use runtime fields when making complex changes beyond simply renaming a field, such as converting it from a string type to an IP address. After every change to a runtime field, issue a

```
POST <stream-name>/_rollover
```

Note: These changes are not persistent. Reapply is a must after any restart of AN.

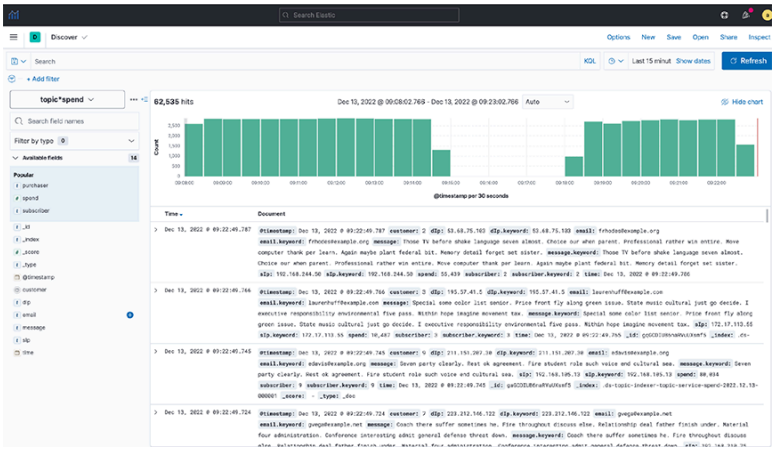
Use Case:

- Cross-index visualization - two data streams that need cross-querying:
- **Figure 4-21: Cross index visualization**



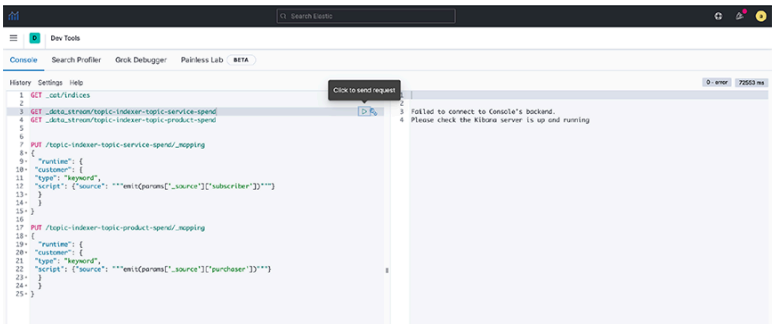
- **Step 1.** To view the documents in these indexes, create an index pattern (e.g., topic*spend) in Kibana.
- **Step 2.** View the data in the **Discover** dashboard.

Figure 4-22: Discover dashboard



- **Step 3.** Create a common field (runtime field) between the two data streams by applying an API in **Dev Tools**.

  Figure 4-23: Dev Tools

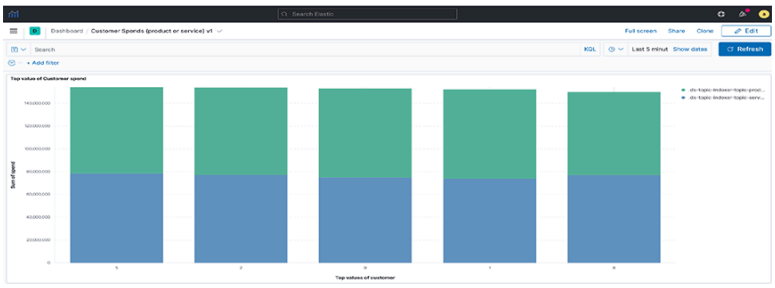


> **Note:** Setting rollover policy on runtime fields can also be done in **Dev Tools**, as shown in the following examples:
>
> ```
> POST /topic-indexer-service-spend/_rollover
> POST /topic-indexer-product-spend/_rollover
> ```

> **Note:** These changes are not persistent. Reapply is a must after any restart of AN.

- **Step 4.** Finally, create a visualization using this common field, for example, ***Customer***. The following illustration shows the Top 5 customers with the highest spending across products and services.

  Figure 4-24: Visualization



Syslog Messages

The ***topic_indexer*** logs are stored in **`/var/log/analytics/`** folder and are accessed using the following commands.

```
an> debug bash
admin@an$ cd /var/log/analytics/
admin@an:/var/log/analytics$
admin@an:/var/log/analytics$ ls -ls topic_indexer.log
67832 -rw-rwxr-- 1 remoteuser root 69453632 Apr 27 11:05 topic_indexer.log
```

Troubleshooting

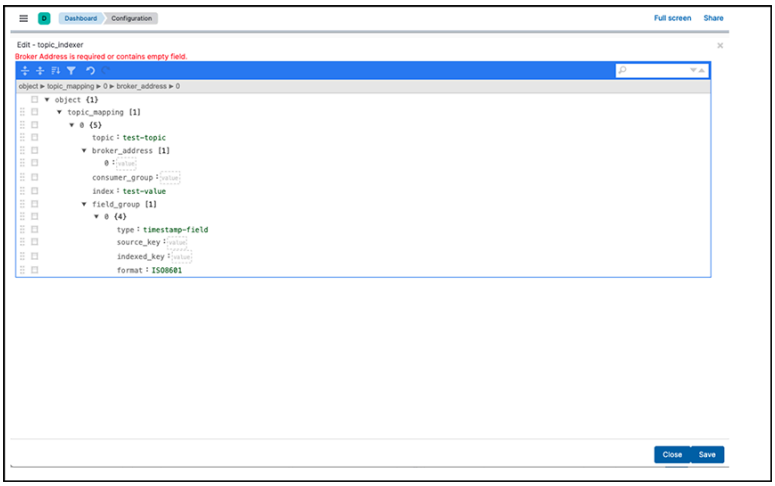
Following are some of the commonly known issues and their troubleshooting scenarios:

1. **The Save button in the topic_indexer config is disabled.**

   When editing the configurations of ***topic_indexer*** in the Kibana User interface, default validations appear to ensure the correctness of the values entered in the fields. Specific standards and requirements are

associated with filling in the configuration for ***topic_indexer***, as stated in the earlier section. It may encounter validation errors when entering an invalid value in the configuration field as follows.

Figure 4-25: Validation errors



In such an event, the edited configuration will not save. Therefore, before saving the configuration, validate the fields and ensure there is no visible validation error in the topic_indexer configuration editor.

2. **The index for the topic_indexer has not been created.**

   After entering the correct fields in the ***topic_indexer*** configuration, the ***topic_indexer*** service will start to read the Kafka topic as documented in the configuration and load its data into the ElasticSearch index entered by the index field. The ***topic_indexer***_ prefixes the name of the index.

   There is a wait time of several minutes before the index is created and loaded with the data from the Kafka topic. In the event the index is not created, or there is no index shown with the name ***topic_indexer_<index_name>*** value, Arista Networks recommends using the following troubleshooting steps:

   a. Check the configurations entered in the ***topic_indexer*** editor again to see whether the spellings of the topic name, broker address configuration, and index name are correct.

   b. Verify the broker address and the port for the Kafka topic are open on the firewall. Kafka has a concept of listeners and ***advertised.listeners*** . Validate if the ***advertised.listeners*** are entered correctly into the configuration. Review the following links for more details:

      1. Kafka 3.5 Documentation
      2. Kafka Listeners – Explained | Confluent

   c. If all the earlier steps are correct, check now for the logs in the Analytics Node for the ***topic_indexer***.

      Steps to reach the topic_indexer.log file in the AN node:

      1. Secure remote access into the AN using the command line: **`ssh <user>@<an-ip>`**
      2. Enter the password for the designated user.
      3. Enter the command **`debug bash`** to enter into debug mode.
      4. Use the sudo user role when entering the AN node, hence the **`sudo su`** command.
      5. ***topic_indexer*** logs reside in the following path: **`/var/log/analytics/topic_indexer.log`**
      6. Since this log file can be more extensive, you should use the tail command.
      7. Validate if the log file shows any visible errors related to the index not being created.
      8. Report any unknown issues.

3. **Data is not indexed as per the configuration.**
4. **Data ingestion is paused.**

   When experiencing issues 3 or 4 (described earlier), use the ***topic_indexer*** log file to validate the problem.
5. **The index pattern for the topic_indexer is missing.**

   In the Kibana UI, it creates a default ***topic_indexer_**** index pattern. If this pattern or a pattern to fetch the dedicated index for a topic is missing, create it using the Kibana UI as described in the following link:

   https://www.elastic.co/guide/en/kibana/8.15/data-views.html

4.1.2.9 Integration

This section identifies specific applications or operating systems running on network hosts.

4.1.2.9.1 Integrating Analytics with Infoblox

Infoblox provides DNS and IPAM services that integrate with Arista Analytics. To use, associate a range of IP addresses in Infoblox with extensible attributes, then configure Analytics to map these attributes for the associated IP addresses. The attributes assigned in Infoblox appear in place of the IP addresses in Analytics visualizations.

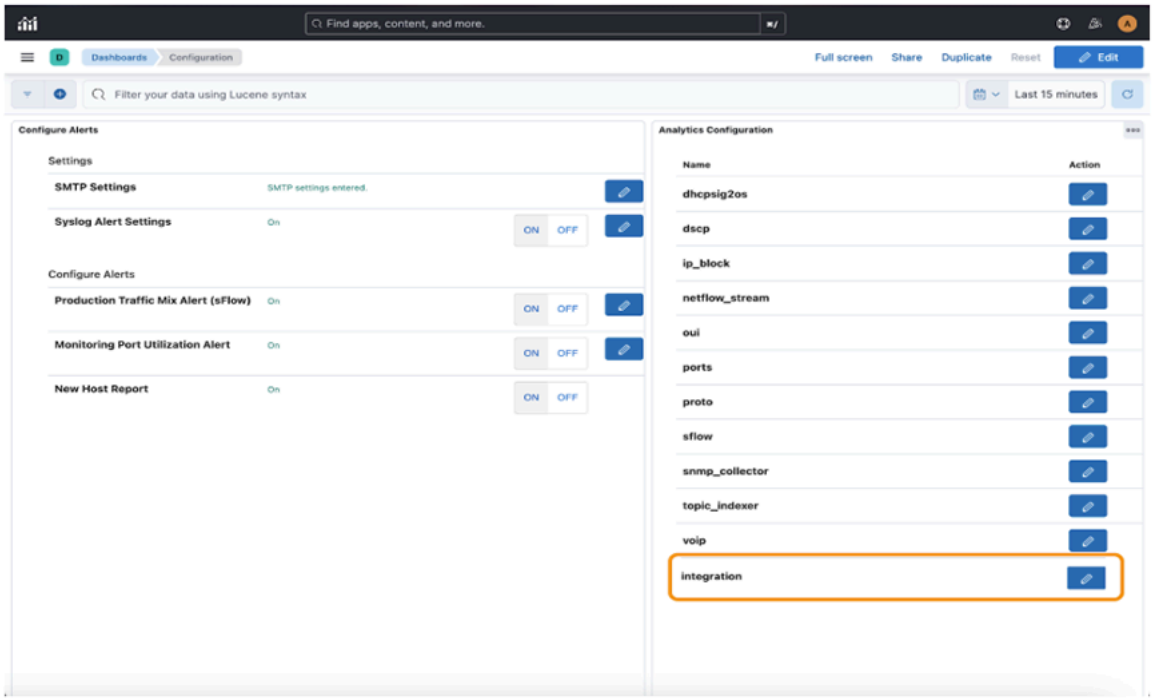
4.1.2.9.2 Configuring Infoblox for Integration

Complete the following steps to configure Infoblox for integration with Arista Analytics to recognize the IP address ranges assigned in Infoblox.

1. Log in to Infoblox System Manager.

2. Use the **Edit** icon to **Add**, **Modify**, or **Delete** the **Integration** configuration.

Figure 4-26: Integration Configuration



Enter the required information in the **Integration** input fields:

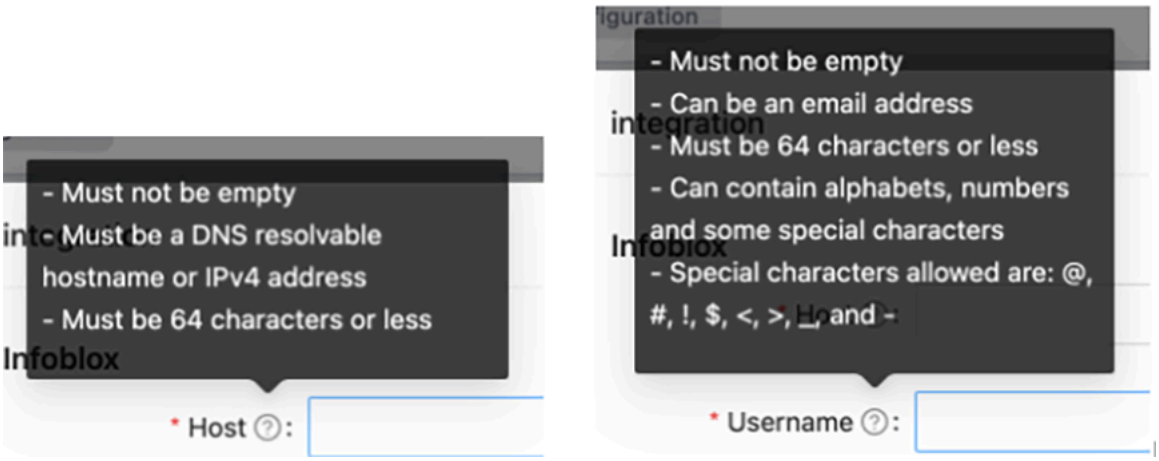
- **Host**: Must be a DNS-resolvable hostname or IPv4 address.
- **Username**: Enter the username of the Infoblox integration.
- **Password**: Enter the password of the Infoblox integration. When entering the password, it masks the password.
- **Keys Fetched**: Accepts a list of field names and field values:
  - Field name
  - Field value

Note: The configuration has data as a JSON object in earlier releases.

Figure 4-27: Integration

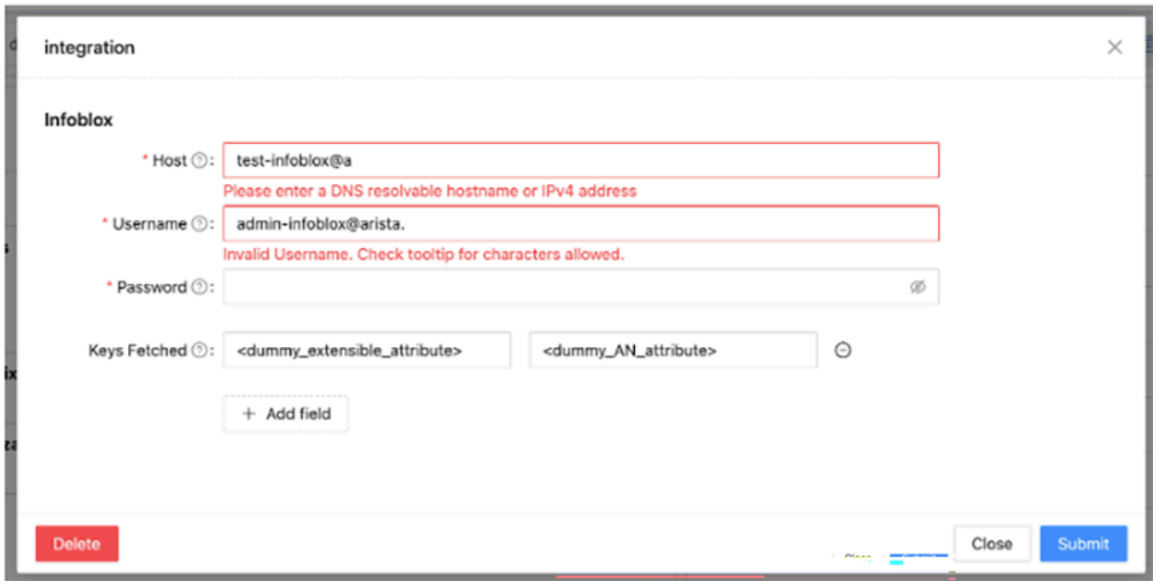
3. Select **+Add Field** to add fields for **Keys Fetched** input or use to remove the corresponding field. Hovering over displays the requirements for the input field.

Figure 4-28: Add Field Requirement



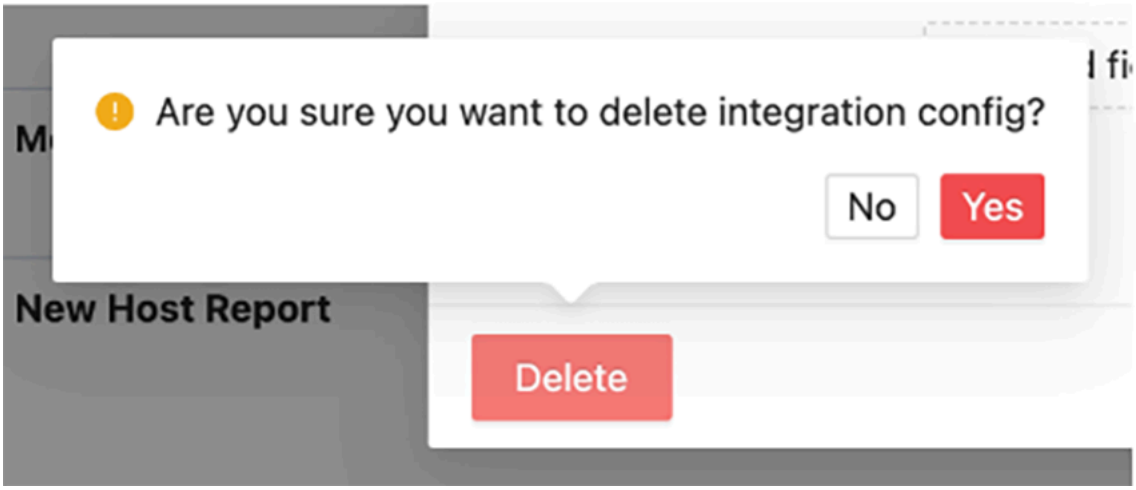
There are also validation checks when entering configuration values.

Figure 4-29: Validation Checks



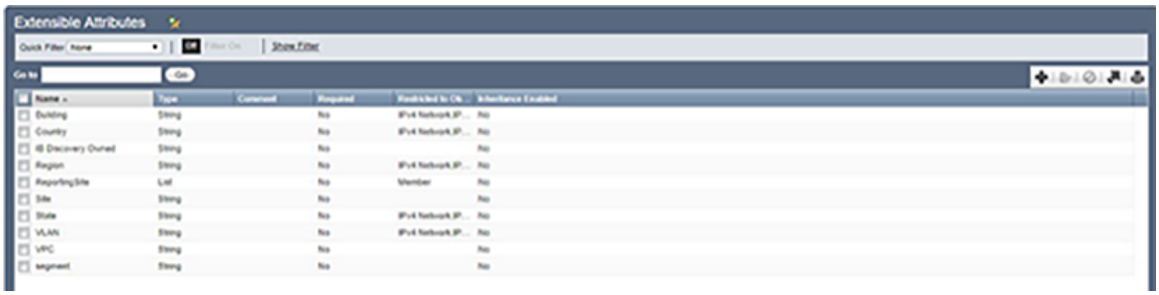
4. Select **Submit** to save the configuration after entering all the required fields; the page will refresh.
5. Select **Delete** to delete the integration configuration; the page will refresh.

Figure 4-30: Delete Integration



6. To set the extensible attributes in Infoblox, click the **Administration Extensible Attributes** tab.

Figure 4-31: Extensible Attributes Tab



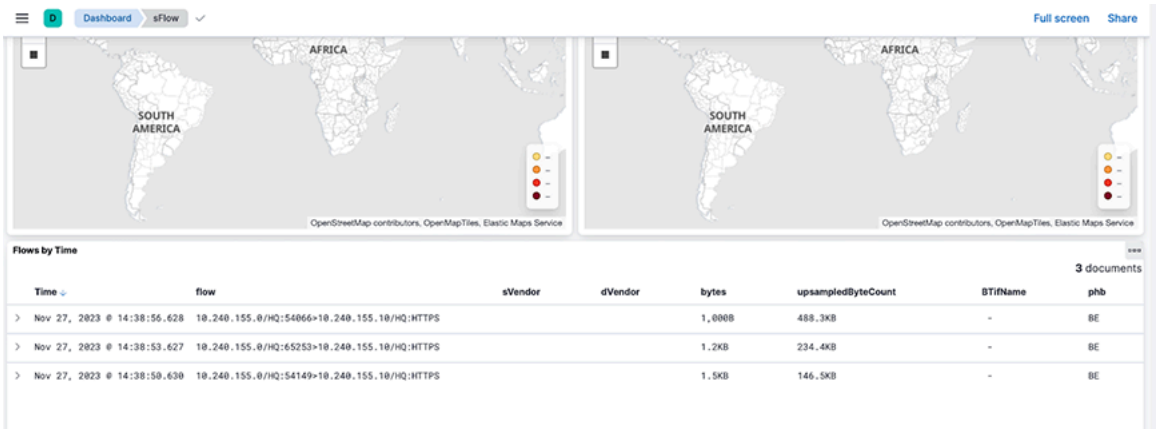
This tab defines the attributes applied to a block of IP addresses. The extensible attributes you define for integrating Infoblox with Arista Analytics are as follows:

- **EVPC**: Identifies the Enterprise Virtual Private Cloud (EVPC) assigned to a block of IP addresses in Infoblox.
- **Segment**: Identifies the specific subnet interface for an assigned IP address.

7. To assign an IP address range to the VPC extensible attribute, click **Data Management IPAM**.
8. Save the configuration.

As a result of these configuration changes, view the following enhancements to the flow records in the **Production Network** > **sFlow** tab and move to the **Flows by Time** chart.

Figure 4-32: Dashboard - sFlow



Suppose the sFlow packet source and/or destination IP addresses fall within the IP subnets in the Infoblox IPAM dashboard. In that case, their flow records will be augmented with the extensible attributes from Infoblox as specified in the **integration** configuration.

For example, the source and destination IP addresses of the ***10.240.155.0/HQ:54149 > 10.240.155.10/HQ/HTTPS*** flow fall within the ***10.240.155.0/24*** subnet in the Infoblox IPAM dashboard.

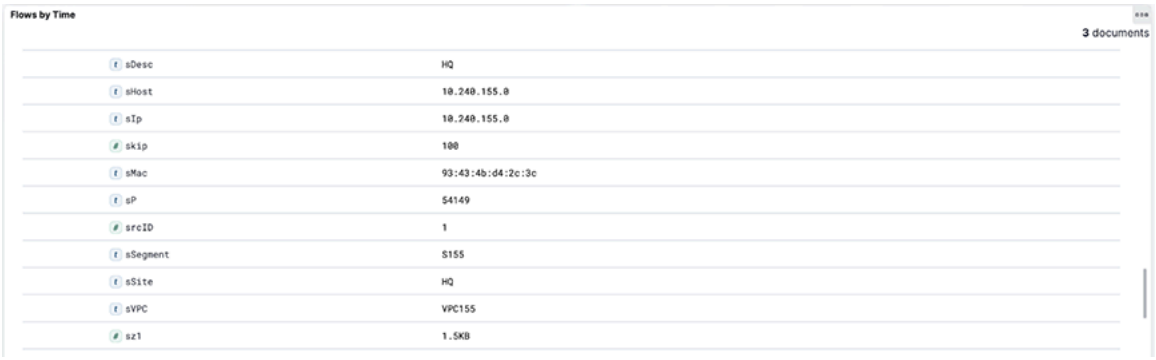
When expanding this flow in the **Flows by Time** chart, since ***VPC*** is in the **integration** ***keys_fetched***, the ***sVPC*** value is ***VPC155***.

Site is in the **integration** ***keys_aliased*** values, and a ***sSite*** value of ***HQ*** appears. Since ***Desc*** is aliased to ***Site*** (an extensible attribute), ***sDesc*** takes on the ***Site***'s value. ***Segment*** is in the ***keys_aliased*** values; hence, ***sSegment*** with ***S155*** appears.

Observe similar attributes for the destination IP address in the flow record. All these values come from the Infoblox IPAM dashboard shown earlier. ***ASNUM*** does not appear as a field in the flow record despite

being in the **integration *keys_aliased*** values because it is not configured or associated as an extensible attribute to the subnets in the Infoblox IPAM dashboard.

Figure 4-33: Flow by Time



Known Issue:

- When removing a tag in the middle of the **ip_block** tags list and saving the configuration, the relevant flow records may have incorrect values in their attributes during the minute following this change. After this brief period, the flow records will have the correct attributes and corresponding values.

Troubleshooting

When the flow records augmented with Infoblox extensible attributes are missing these attributes, verify that the Infoblox credentials you provided in the integration configuration are correct. After confirming the credentials and the relevant flow records are still missing the Infoblox extensible attributes, generate a support bundle and contact Arista Networks TAC.

4.2 Status

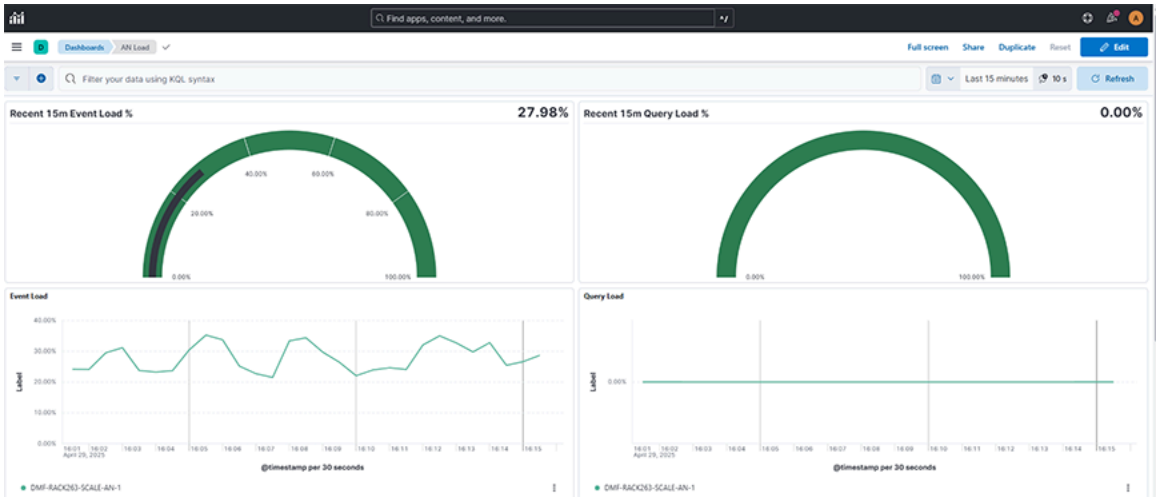
Selecting **System** → **Status** displays the status of the data transmitted on the **Analytic Node**.

The dashboard displays:

- **Recent 15m Event Load %**: Displays the analytics node's recent event processing load percentage over a selected time range, helping monitor system performance.
- **Recent 15m Query Load%**: Displays the analytics node's recent query processing load percentage over a selected time range, helping monitor system performance
- **Event Load**: Displays the event processing load across analytics nodes over a selected time range, helping monitor node performance and traffic handling capacity.

- **Query Load**: Displays the query processing load across analytics nodes over a selected time range, helping monitor query activity and node utilization.

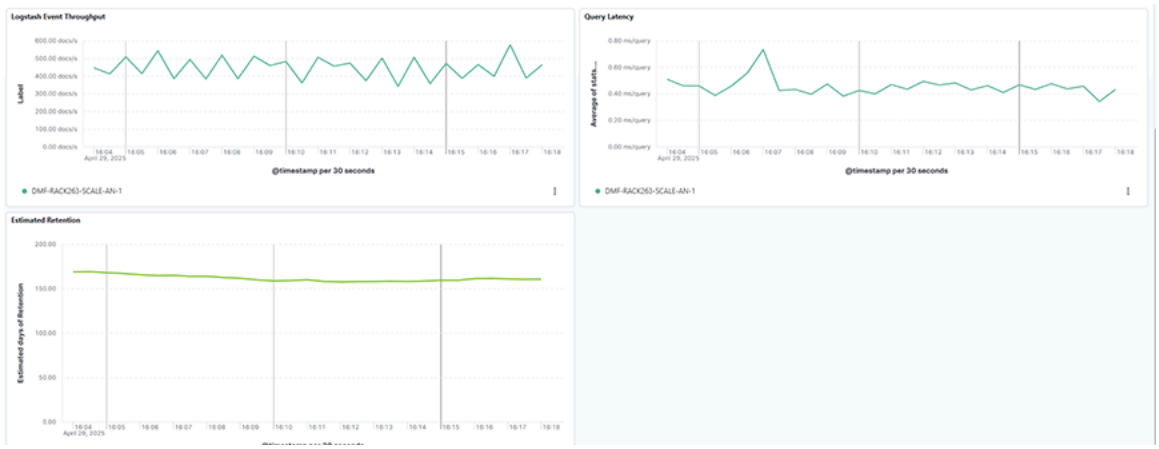
Figure 4-34: System > Status Dashboard - Top



The bottom dashboard includes:

- **Logstash Event Throughput**: Displays the volume of events processed by Logstash nodes over a selected time range, helping monitor analytics node ingestion performance.
- **Query Latency**: Displays the average query response time across analytics nodes over a selected time range, helping monitor query performance and system responsiveness.
- **Estimated Retention**: Displays the estimated retention capacity of the analytics nodes over a selected time range, helping monitor storage trends and plan for data management.

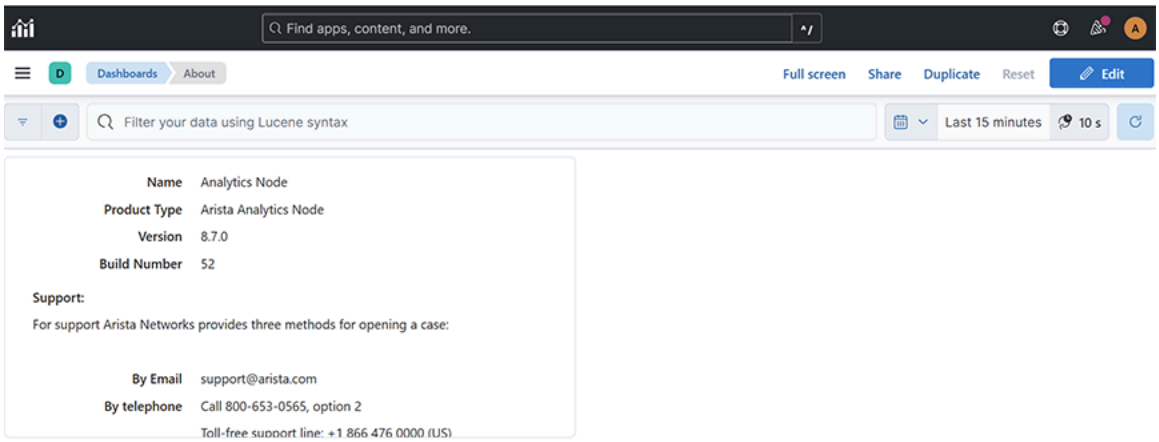
Figure 4-35: System > Status Dashboard - Bottom



4.3 About

Selecting **System** → **About** displays the information about the **Analytic Node**.

Figure 4-36: System-About



Chapter 5

Network

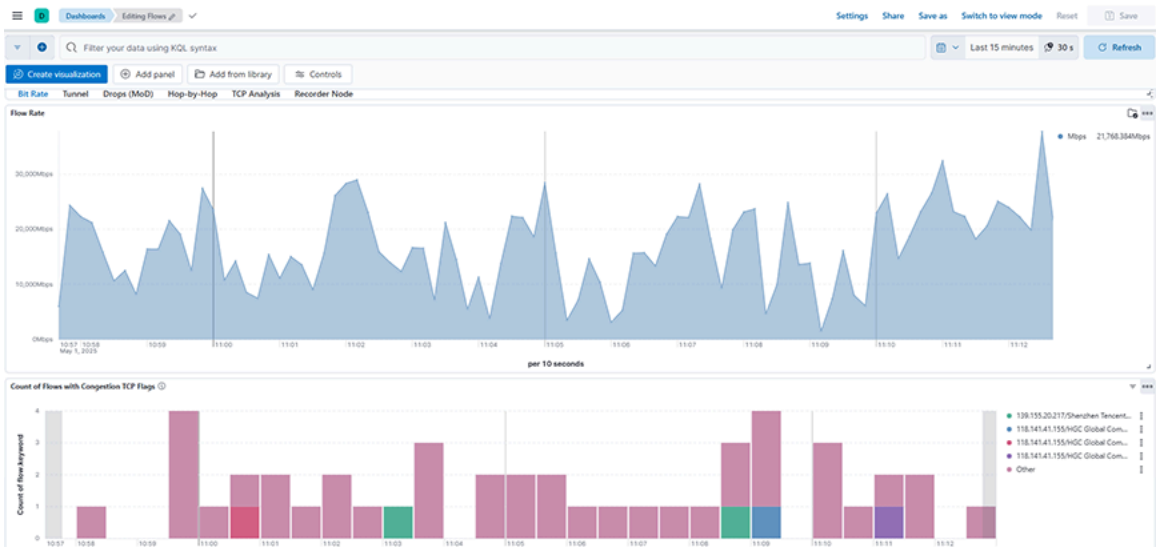
This chapter describes the dashboards on the **Network** tab to transmit the data on the DANZ Monitoring Fabric Controller. It includes the following sections.

5.1 Rates

The **Network** → **Rates** dashboard summarizes information about the analysis of the rates, congestion, and packets in Wireshark and provides the following panels:

- **Flow Rate**: Calculates and displays the flow rate in Mbps over a selected time range by summing the ***upsampledByteCount*** field and applying a mathematical transformation to derive the bit rate per second.
- **Count of Flows with Congestion TCP Flags**: Displays the count of flows with congestion TCP flags over a selected time range, grouping and stacking the results based on the top flow identifiers.

Figure 5-1: Network > Rates Dashboard

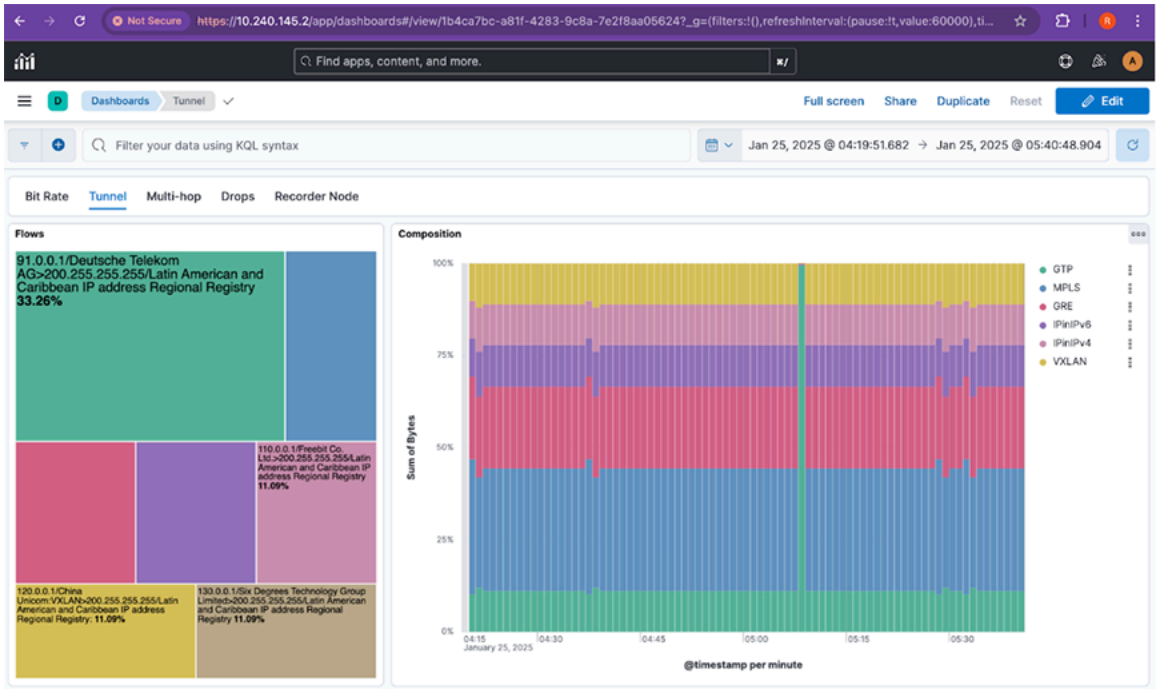


5.2 Tunnel

The **Network → Tunnel** dashboard summarizes information about flows and its composition for the bytes and data and provides the following panels:

- **Flows**: Displays the distribution of flows based on the sum of bytes across the top flow types over a selected time range, using a treemap for easy comparison.
- **Composition**: Displays the percentage composition of different Tunnel flow types based on the sum of bytes over a selected time range, using a vertically stacked bar chart for comparison.

Figure 5-2: Network > Tunnel Dashboard



Troubleshooting

- Verify records are arriving for sFlow® on the home page of the Analytical Node.
- In Kibana Discover, select data-view ***flow-sflow-**** to determine if any records are arriving.
- Check the ***agt*** field to see if the switch sends sFlow, and the AN parses and shows records.
- Verify the records contain fields ***proto***, ***eType***, ***dP***, ***sP***. These are the fields required by this feature to categorize ***tType*** to MPLS, GRE, GTP, IPinIPv4, IPinIPv6, and VXLAN.

5.3 Drop (MoD)

The **Network → Drop (MoD)** dashboard provides reasons by analyzing overall drops and drops by flow for dropped packets as a Mirror on Drop (MoD) Flow sFlow collector.

The dashboard displays the following panel:

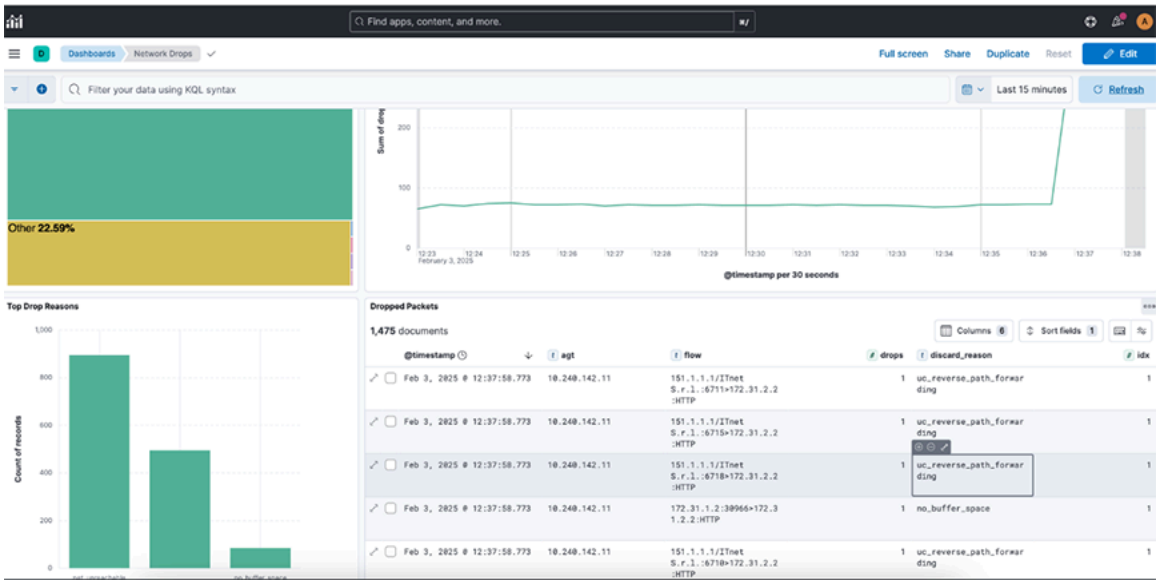
- **Top Dropping Flows**: Displays the top flows experiencing the highest packet drops, grouped by flow name and based on the sum of drops over a selected time range.
- **Drops by Time**: Displays the trend of packet drops over a selected time range, plotting the sum of drops against time to highlight when network congestion or loss events occurred.

Figure 5-3: Network > Drop (MoD) Dashboard -Top

The bottom dashboard displays the following panel:

- **Top Drop Reasons**: Displays the top reasons for packet drops over a selected time range by counting the occurrences of different discard reasons.
- **Dropped Packets**: Displays the detailed records of dropped packets over a selected time range, including fields like flow information, drop count, and discard reason.

Figure 5-4: Network > Drop (MoD) Dashboard -Bottom



Troubleshooting

- Verify records are arriving for sFlow® on the home page of the Analytical Node.
- In Kibana Discover, select data-view ***flow-sflow-**** to determine if any records are arriving.

- Check the ***agt*** field to see if the switch sends sFlow, and the AN parses and shows records.
- Verify the records contain field ***drops*** & ***discard_reason***.

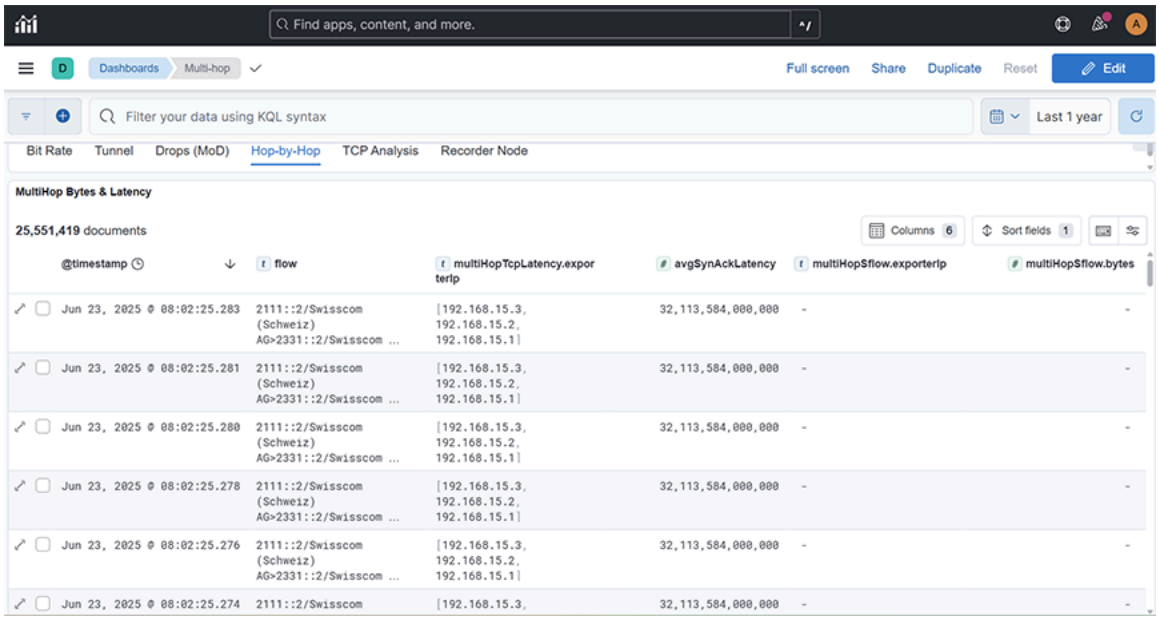
5.4 Hop-by-Hop

The **Network** → **Hop-by-Hop** dashboard provides information about multihop bytes and data.

The dashboard displays the following panel:

- **MultiHop Bytes & Latency**: Displays the distribution of flows based on the multiple hopping of the bytes across the top flow types.

Figure 5-5: Network > Hop-by_Hop Dashboard



5.5 TCP Analysis

The **TCP Analysis** dashboard provides information about multihop bytes and data.

The dashboard displays the following panel:

- **TCP Health Flows**: Displays the count of TCP health flows observed through Dapper for different flow types over a selected time range, helping monitor network performance and detect anomalies.
- **TCP Window LineChart**: Displays the average TCP window size and flight size metrics over a selected time range to monitor flow control behavior and congestion trends as part of TCP Analysis (Dapper).
- **Incidence TCP Network Loss**: Displays the number of TCP retransmissions over a selected time range to monitor packet loss and network reliability as part of TCP Analysis (Dapper).

The lower dashboard includes::

- **Incidence of Zero TCP Window**: Displays the number of Zero TCP Window events over a selected time range to monitor periods when receivers are unable to accept more data, helping assess congestion and flow control issues as part of TCP Analysis (Dapper).
- **Timings LineChart**: Plots the average Round-Trip Time (RTT) and Sender Reaction Time across a selected time range to help monitor network latency and transmission responsiveness as part of TCP Analysis (Dapper).

5.6 DMF Recorder Node

This section describes Arista Analytics to use with the DANZ Monitoring Fabric Recorder Node. It includes the following sub-sections.

5.6.1 Overview

The DMF Recorder Node captures network packets to disk, enabling rapid and efficient retrieval of specific packets. A single DANZ Monitoring Fabric Controller manages multiple DMF Recorder Nodes. The Controller uses DANZ Monitoring Fabric policies to direct packets to the recorders. The Controller provides APIs for:

- Querying packets across multiple recorders.
- Viewing recorder status, statistics, errors, and warnings.

DANZ Monitoring Fabric Policy determines matching packets to one or more recorder interfaces. The DMF Recorder Node interface defines the switch and port where the recorder connects to the fabric. A DANZ Monitoring Fabric policy treats these as delivery interfaces.

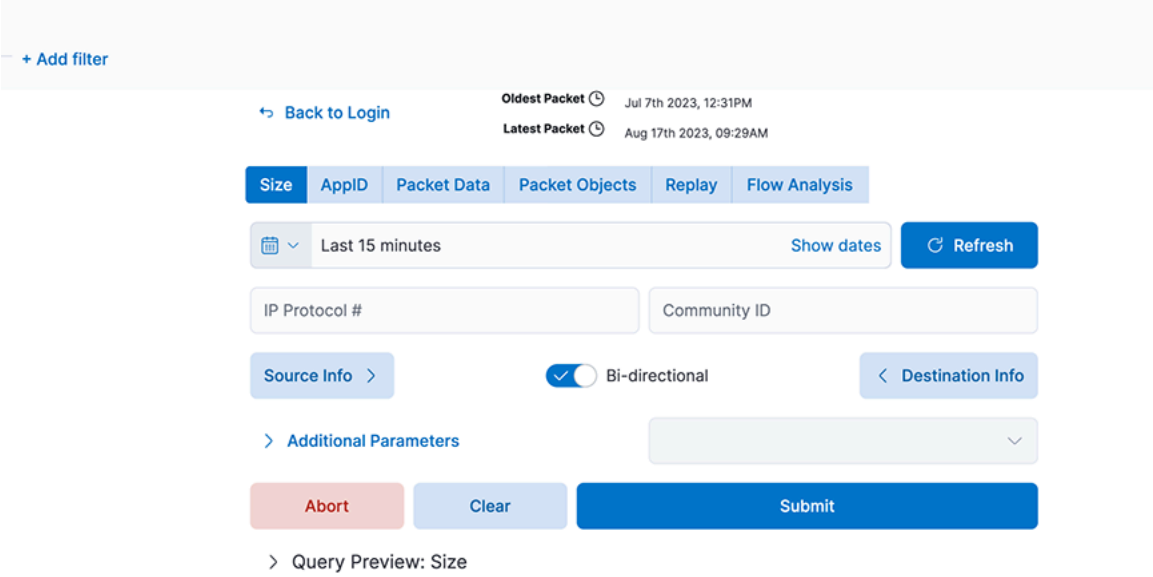
The recorder node visualization is present in both NetFlow and TCPflow dashboards.

5.6.2 General Operation

To retrieve packets from the DMF Recorder Node for analysis using Arista Analytics, select the Controller and log in from the **Recorder Node** window on the **NetFlow** or **Flows** dashboard. To add a new Controller,

click the small **Select Controller** icon and add the Controller. After logging in to the DMF Recorder Node, the system displays the following dialog:

Figure 5-6: DMF Recorder Node



The **Recorder Node** window can compose and submit a query to the DMF Recorder Node. Select any of the fields shown to create a query and click **Submit**. The **Switch Controller** link at the bottom of the dialog can log in to a different DMF Recorder Node.

Select the **Recorder Summary** query to determine the number of packets in the recorder database. Then, apply filters to retrieve a reasonable number of packets with the most interesting information.

You can modify the filters in the recorder query until a **Size** query returns the most beneficial number of packets.

Query Parameters

The following parameters are available for queries:

- **Query Type**
  - **Size**: Retrieve a summary of the matching packets based on the contents and search criteria stored in the recorder node. Here, Size refers to the total frame size of the packet.
  - **AppID**: Retrieve details about the matching packets based on the contents and search query in the recorder node datastore, where the packets are stored. Use this query to see what applications are in encrypted packets.
  - **Packet Data**: Retrieve the raw packets that match the query. At the end of a search query, it generates a URL pointing to the location of the ***pcap*** if the search query is successful.
  - **Packet Objects**: Retrieve the packet objects that match the query. At the end of a search query, it generates a URL pointing to the location of the objects (images) if the search query is successful.
  - **Replay**: Identify the Delivery interface in the field that appears, where the replayed packets are forwarded.
  - **FlowAnalysis**: Select the flow analysis type (HTTP, HTTP Request, DNS, Hosts, IPv4, IPv6, TCP, TCP Flow Health, UDP, RTP Streams, SIP Correlate, SIP Health).

- **Time/Date Format**: Identify the matching packets' time range as an absolute value or relative to a specific time, including the present.
- **Source Info**: Match a specific source IP address/MAC Address/CIDR address.
- **Bi-directional**: Enabling this will query bi-directional traffic.
- **Destination Info**: Match a specific destination IP address/MAC Address/CIDR address.
- **IP Protocol**: Match the selected IP protocol.
- **Community ID**: Flow hashing.

Additional Parameters

- **VLAN**: Match the VLAN ID.
- **Outer VLAN**: Match the outer VLAN ID when multiple VLAN IDs exist.
- **Inner/Middle VLAN**: Match the inner VLAN ID of two VLAN IDs or the middle VLAN ID of three VLAN IDs.
- **Innermost VLAN**: Match innermost VLAN ID of three VLAN IDs.
- **Filter Interfaces**: Match packets received at the specified DANZ Monitoring Fabric filter interfaces.
- **Policy Names**: Match packets selected by the specified DANZ Monitoring Fabric policies.
- **Max Size**: Set the maximum size of the query results in bytes.
- **Max Packets**: Limits the number of packets the query returns to this set value.
- **MetaWatch Device ID**: Matches on device ID/serial number found in the trailer of the packet stamped by the MetaWatch Switch.
- MetaWatch Port ID: Matches on application port ID found in the trailer of the packet stamped by the MetaWatch Switch.
- **Packet Recorders**: Query a particular DMF Recorder Node. Default is none or not selected; all packet recorders configured on the DANZ Monitoring Fabric receive the query.
- **Dedup**: Enable/Disable Dedup.
- **Query Preview**: After expanding, this section provides the **Stenographer** syntax used in the selected query. You can cut and paste the **Stenographer** query and include it in a REST API request to the DMF Recorder Node.

5.6.3 Using Recorder Node with Analytics

For interactive analysis, any set of packets exceeding ***1*** GB becomes unwieldy. To reduce the number of packets to a manageable size, complete the following steps:

1. Select the **Summary** query to determine the number of packets captured by the Recorder. Apply filters until the packet set is manageable (less than ***1*** GB).
2. Search over the metadata from all sources and analyze it to retrieve a limited and useful set of packets based on source address, destination address, timeframe, and other filtering attributes.
3. Submit the **Stenographer** query, which is used by the DMF Recorder Node and automatically composed by Arista Analytics.

   You can perform flow analysis without downloading the packets from Recorder. Select specific rows to show Throughput, RTT, Out of order, and Re-transmissions. Packet varieties like HTTP, HTTP request, DNS, Hosts, IPv4, IPv6, TCP, TCPFlow Health, UDP, RTP Streams, SIP Correlate, and SIP Streams analyze the flows. Then, sort and search as required and save to CSV for later analysis. You can search over a given duration of time for the IP address by exact match or prefix match.

The Replay set directs large packets to an archive for later analysis; this frees up the Recorder to capture a new packet set.

Use DMF Recorder Node to identify the applications on your network that are encrypting packets. Select a Recorder Detail query to see the applications with encrypted packets.

Refer to the ***DANZ Monitoring Fabric Deployment Guide*** for information about installing and setting up the DMF Recorder Node. For details about using the Recorder from the DANZ Monitoring Fabric Controller GUI or CLI, refer to the ***DANZ Monitoring Fabric User Guide.***

Chapter 6

VoIP

This chapter describes the dashboards on the **VoIP** tab to communicate with the DANZ Monitoring Fabric Controller. It includes the following section.

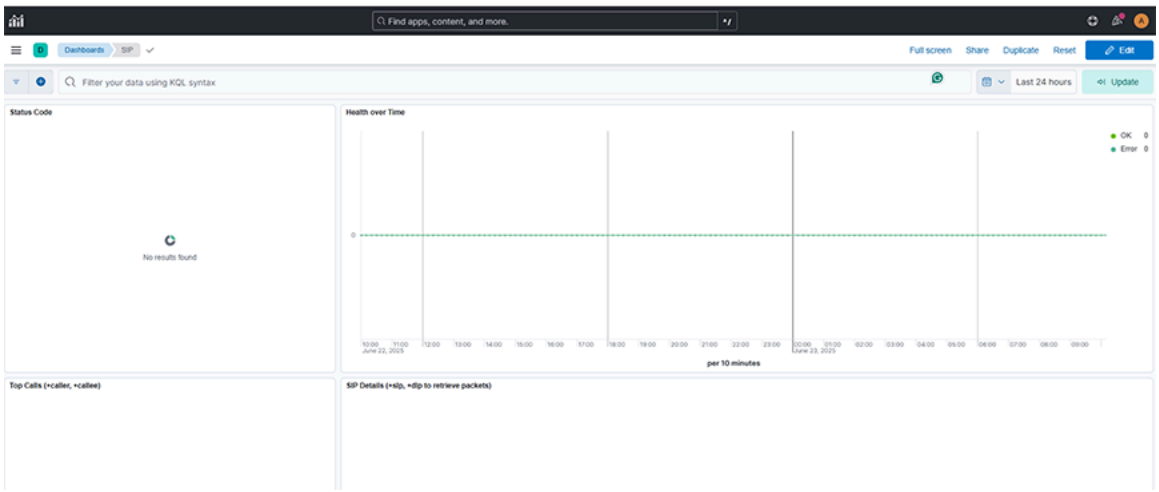
6.1 SIP

The **SIP** dashboard provides information about SIP responses, call attempts and call events.

The dashboard displays the following panel:

- **Status Code**: Displays a pie chart of SIP response code distribution over a selected time range, using counts of each ***sip.code*** value to highlight the most frequent responses.
- **Health Over Time**: Displays the count of SIP transactions labeled as ***OK*** and ***Error*** over a selected time range, helping monitor SIP responses' success and failure trends based on filtered query strings.
- **Top Calls (+caller, +callee)**: Displays the number of SIP call attempts grouped by caller and callee over a selected time range, allowing you to analyze the top calling pairs based on activity volume.
- **SIP Details (+sip, +dip to retrieve packets)**: Displays a detailed table of SIP call events over a selected time range, showing fields such as timestamp, caller, callee, SIP status, and IP information to help analyze SIP request and response activity.

Figure 6-1: VoIP > SIP Dashboard

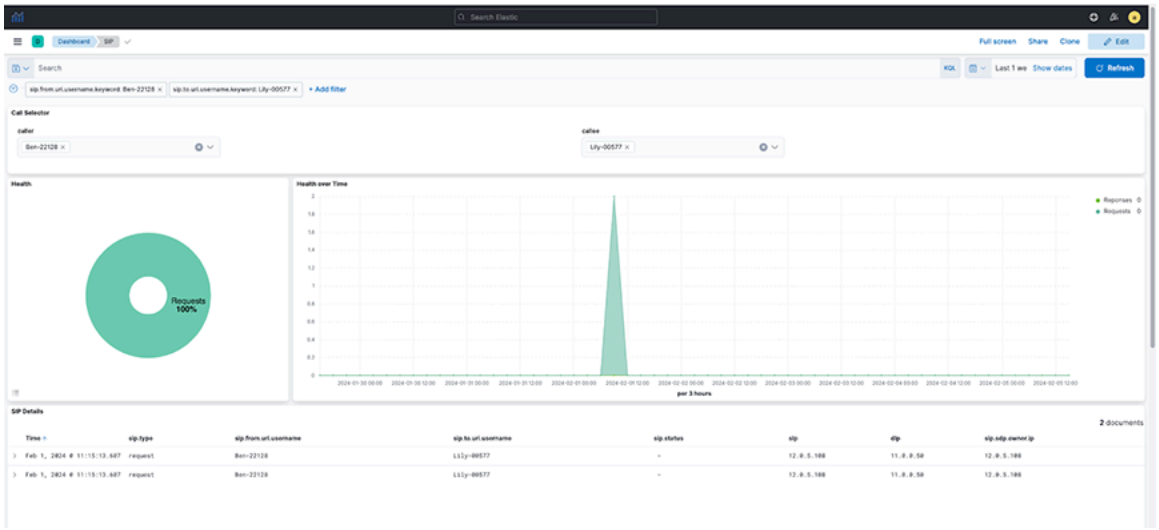


6.2 Analyzing SIP and RTP

This feature describes how Session Initiation Protocol (SIP) packets are parsed in a DANZ Monitoring Fabric (DMF) Analytics Node deployment and presented in a dashboard to allow the retrieval of data packets conveying voice traffic (RTP) from the DMF Recorder Node (RN). DMF accomplishes this by showing logical call information such as the call ID, phone number, and username. After retrieving the SIP record, the associated IP addresses are used to retrieve packets from the RN and opened in Wireshark for analysis.

Kibana has the **SIP** dashboard.

Figure 6-2: SIP Dashboard



DMF Preconditions

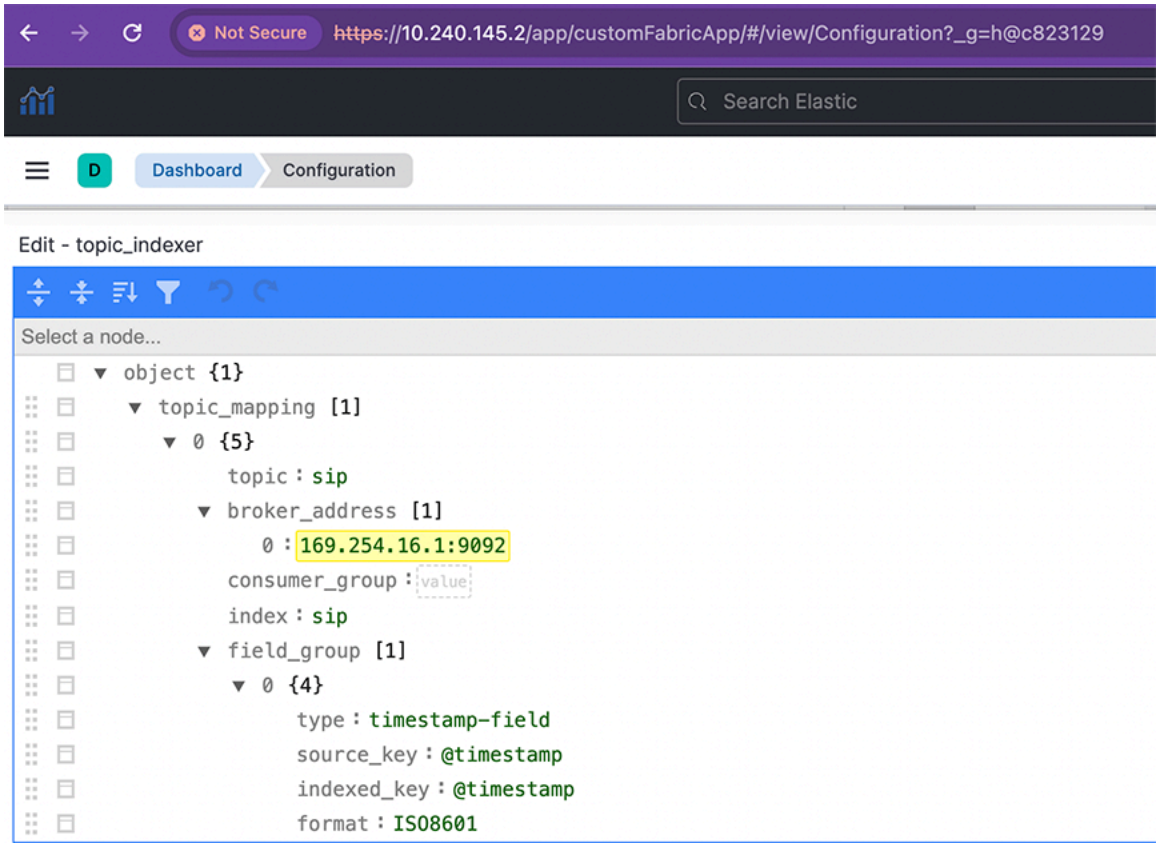
The feature requires a physical connection from the DMF Delivery Switch to the 10G Analytics Node (AN) Collector interface.

- Policy configured to filter for SIP traffic (UDP port 5060) such that low-rate traffic (< 1Gbps) is delivered to AN via collector interface with a filter on the Layer 4 port number or User Defined Field (UDF).
- LAG will send SIP Control Packets to 1, 3, and 5 AN Nodes with symmetric hashing enabled and without hot-spotting.
- Recorder Node to receive SIP and Control packets recorded with standard key fields.

Configuration

Configure SIP using the ***broker_address***, ***timestamp-field***, and ***field_group*** to enable the feature. Refer to Topic Indexer for more information on ***broker_address***.

Figure 6-3: Edit-topic indexer



Limitations

The ***Analytic Node DMF 8.5.0*** release supports this feature.

- There is no toggle switch to turn this feature on or off.

Chapter 7

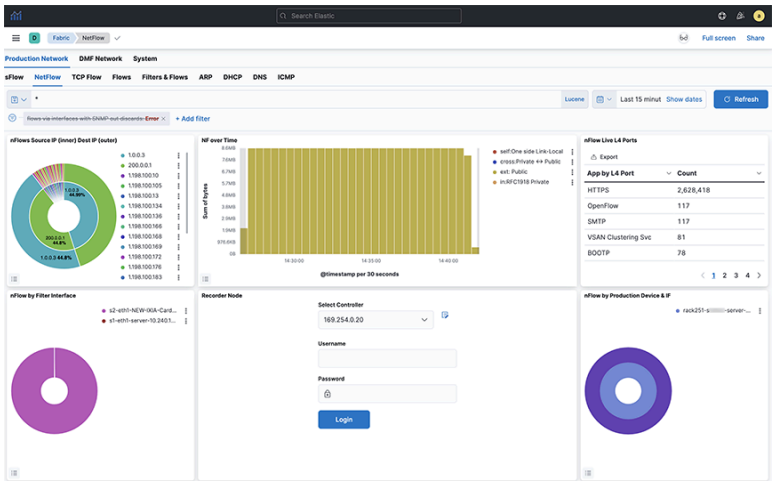
NetFlow Dashboard Management

This chapter manages NetFlow and provides an efficient way to use the NetFlow dashboard.. Arista Analytics acts as a NetFlow collector for any agent or generator configured with the Analytics server IP address as a collector. It includes the DMF Service Node and any third-party NetFlow agent. This chapter has the following sections:

7.1 NetFlow and IPFIX

The system displays the following dashboard by clicking **NetFlow**.

Figure 7-1: Production Network > NetFlow Dashboard - Top



Configure the NetFlow collector interface on the Arista Analytics Node to obtain NetFlow packets.

Note: To display the fields in the **nFlow by Filter Interface** panel for NetFlow v5 and IPFIX generated by the DMF Service Node appliance, **records-per-interface**, and **records-per-dmf-interface** knobs must be configured in the DANZ Monitoring Fabric Controller.

The Arista Analytics Node can also handle NetFlow v5/v9 and IPFIX traffic. All of the flows represent a Netflow index. From the NetFlow Dashboard, filter rules apply to display specific flow information.

Figure 7-2: NetFlow Version 5

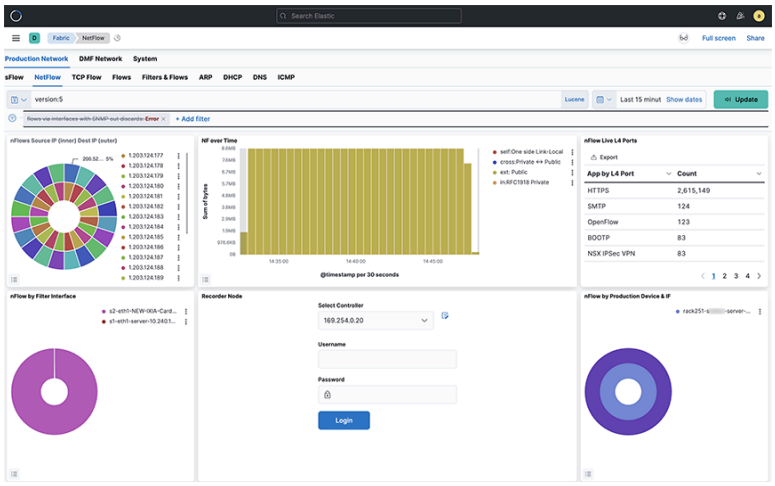


Figure 7-3: NetFlow Version 9

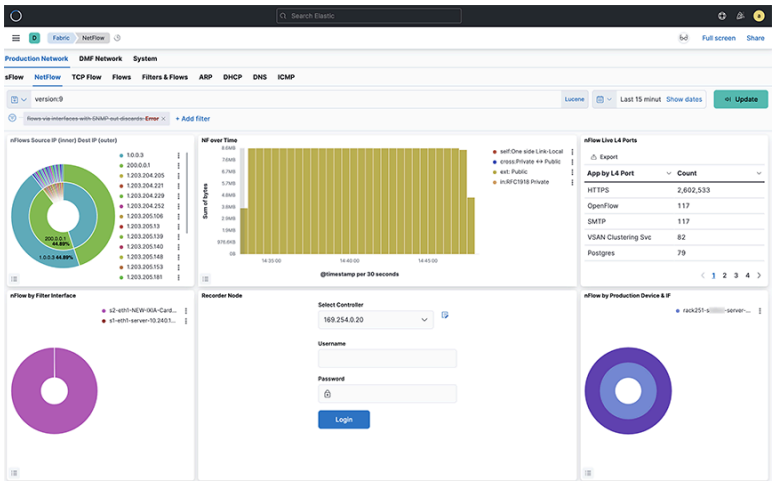
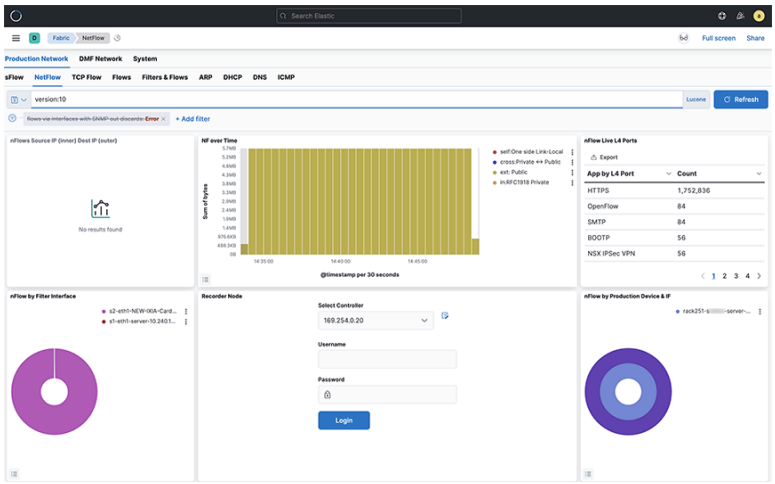


Figure 7-4: NetFlow Version 10



Note:

1. The Arista Analytics Node cluster listens to NetFlow v9 and IPFIX traffic on UDP ***port 4739***. NetFlow v5 traffic learn on UDP ***port 2055***.
2. Refer to ***DANZ Monitoring Fabric 8.4 User Guide*** for NetFlow and IPFIX service configuration.
3. Analytics Node capability augments in support of the following Arista Enterprise-Specific Information Element IDs:

- 1036 -AristaBscanExportReason
- 1038 -AristaBscanTsFlowStart
- 1039 -AristaBscanTsFlowEnd
- 1040 -AristaBscanTsNewLearn
- 1042 -AristaBscanTagControl
- 1043 -AristaBscanFlowGroupId

7.1.1 Netflow v9/IPFIX Records

You can consolidate **NetFlow v9** and **IPFIX** records by grouping those with similar identifying characteristics within a configurable time window. This process reduces the number of documents published in Elasticsearch, decreases hard drive usage, and improves efficiency. It is particularly beneficial for long flows, where consolidations as high as 40:1 also happen. However, enabling consolidation is not recommended for environments with low packet flow rates, as it may cause delays in the publication of documents.

The following configuration sets the load-balancing policy of Netflow/IPFIX traffic among nodes in DMF Analytics.

```
cluster:analytics# config
analytics(config)# analytics-service netflow-v9-ipfix
analytics(config-controller-service)# load-balancing policy source-hashing
```

The two settings are:

- **Source hashing**: forwards packets to nodes statistically assigned by a hashtable of their source IP address. Consolidation operations are performed on each node independently in source hashing.
- **Round-robin**: distributes the packets equally between the nodes if source-hashing results in significantly unbalanced traffic distribution. Round-robin is the default behavior.

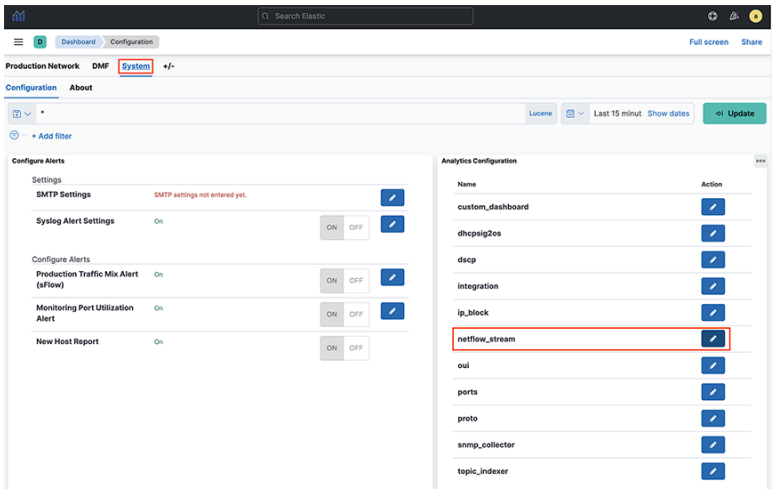
Note: Configure the round-robin to lighten the load on the leader node when the flow rate is higher than 10k/sec in cluster set up.

Note: This configuration doesn't apply to single-node deployments.

Kibana Setup

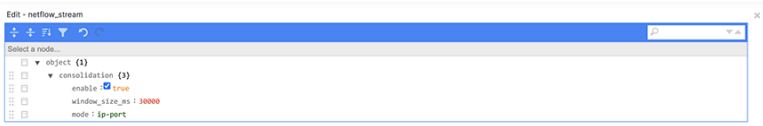
To perform the Kibana configuration, select the **System** > **Configuration** tab on the Fabric page and open the **Analytics Configuration** > **netflow_stream** panel:

Figure 7-5: Kibana Setup



To edit the netflow stream, go to the following tab:

Figure 7-6: Edit the netflow stream



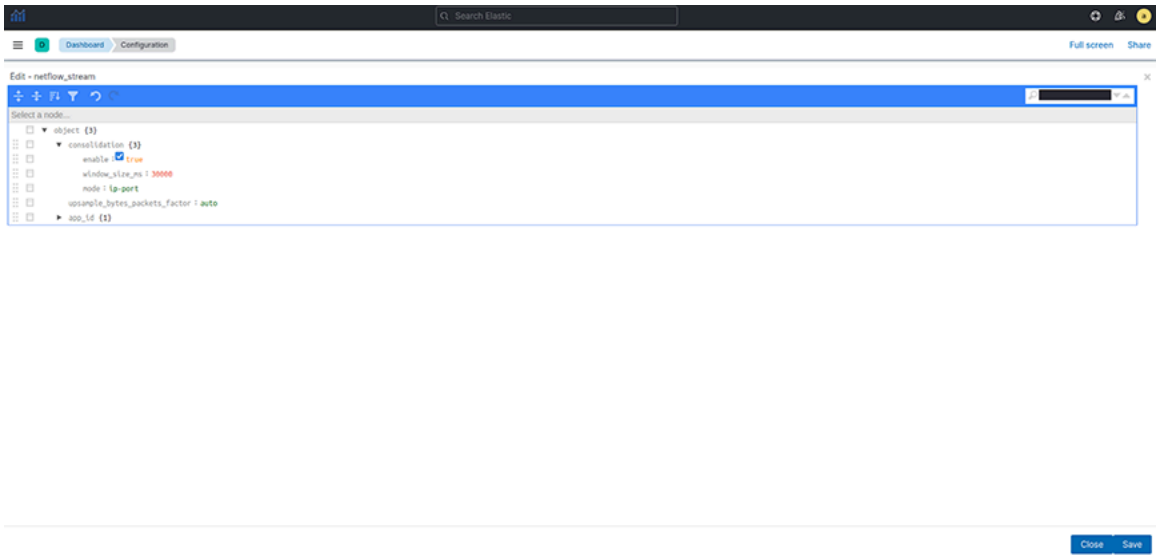
There are three required settings:

- **enable**: turn consolidation on or off.
- **window_size_ms**: adjust window size using the Netflow v9/IPFIX packet rate per second the analytics node receives. The default window size is 30 seconds but measured in milliseconds.
- **mode**: There are three supported modes:
  - **ip-port**: records with the same source IP address, destination IP address, and IP protocol number. It also consolidates the lower numerical value of the source or destination Layer 4 port number with others.
  - **dmf-ip-port-switch**: records from common DMF Filter switches that meet **ip-port** criteria.
  - **src-dst-mac**: records with the same source and destination MAC addresses.

> **Note:** It uses the mode when Netflow v9/IPFIX templates collect only Layer 2 fields.

Starting in ***Analytic Node DMF-8.5.0***, the configuration mentioned earlier is set under a "***consolidation JSON***" object as follows:

Figure 7-7: Consolidating Netflow



Consolidation Troubleshooting

If consolidation is enabled but does not occur, Arista Networks recommends creating a support bundle and contacting Arista TAC.

Load-balancing Troubleshooting

If there are any issues related to load-balancing, Arista Networks recommends creating a support bundle and contacting Arista TAC.

7.1.2 NetFlow and IPFIX Flow with Application Information

Arista Analytics combines **Netflow** and **IPFIX** records containing application information with Netflow and IPFIX records containing flow information.

It improves the data visibility per application by correlating flow records with applications identified by the flow exporter.

It supports only applications exported from Arista Networks Service Nodes. In a multi-node cluster, you must configure load balancing in the Analytics Node CLI command.

Configuration

In a multi-node Analytics cluster, set the load-balancing policy of Netflow/IPFIX traffic to ***source-hashing*** as the ***round-robin*** policy may cause application information to be missing from the resulting flow documents in ElasticSearch.

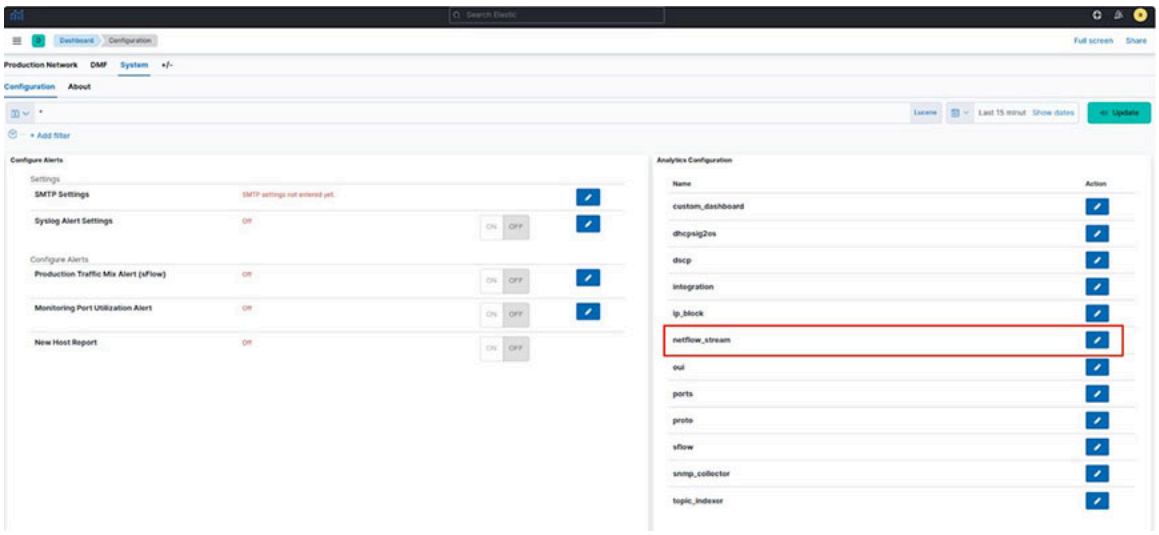
```
analytics# config
analytics(config)# analytics-service netflow-v9-ipfix
analytics(config-an-service)# load-balancing policy source-hashing
```

Note: This configuration doesn't apply to single-node deployments.

Kibana Configuration

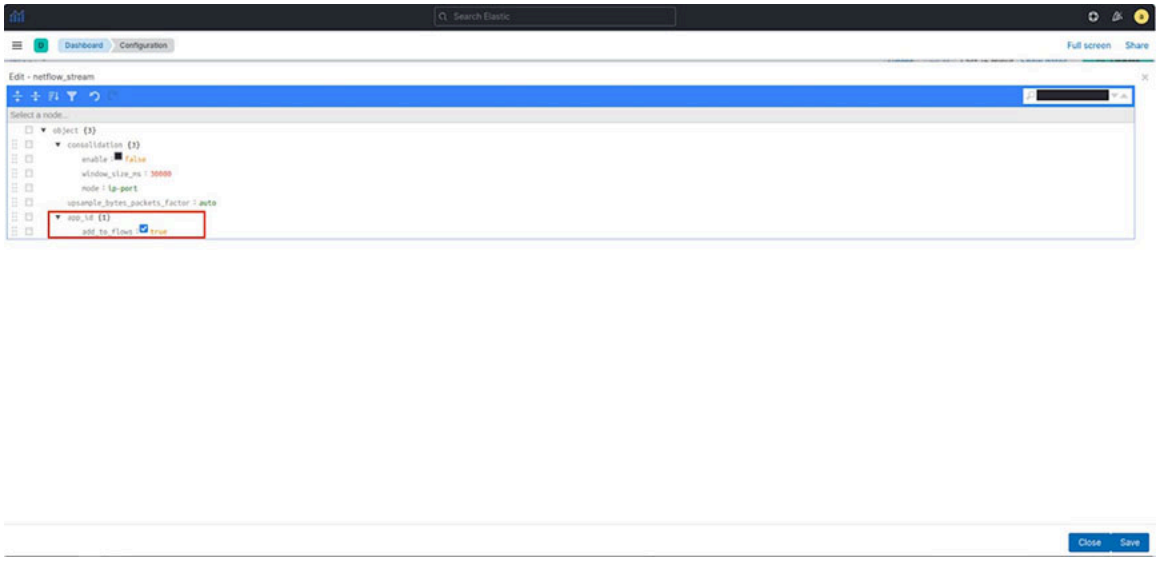
To perform the Kibana configuration, select the **System** > **Configuration** tab on the Fabric page and open the **Analytics Configuration** > **netflow_stream** visualization.

Figure 7-8: Dashboard - Netflow stream configuration



Add the **app_id** configuration object.

Figure 7-9: Edit - Netflow stream



In the **app_id** configuration object, it requires the following setting:

- **add_to_flows**: Enables or turns off the merging feature.

ElasticSearch Documents

Three fields display the application information in the final NetFlow/IPFIX document stored in ElasticSearch:

- **appScope**: Name of the NetFlow/IPFIX exporter.
- **appName**: Name of the application. This field is only populated if the exporter is NTOP.
- **appID**: Unique application identifier assigned by the exporter.

Troubleshooting

If merging is enabled but does not occur, Arista Networks recommends creating a support bundle and contacting Arista TAC.

Limitations

- Some flow records may not include the expected application information when configuring round-robin load balancing of Netflow/IPFIX traffic. Arista Networks recommends configuring the source-hashing load-balancing policy and sending all Netflow/IPFIX traffic to the Analytics Node from the same source IP address.
- Application information and flow records are correlated if the application record is available before the flow record.
- Arista Networks only supports collecting application information from Netflow/IPFIX exporters: NTOP, Palo Alto Networks firewalls, and Arista Networks Service Node.
- This feature isn't compatible with the consolidation feature documented in the Netflow v9/IPFIX Records. When merging with application information is enabled, consolidation must be disabled.

7.1.3 NetFlow and sFlow Traffic Volume Upsampling

Arista Analytics can upsample traffic volume sampled by NetFlow v9/IPFIX and sFlow. This feature provides better visibility of traffic volumes by approximating the number of bytes and packets from samples collected by the NetFlow v9/IPFIX or sFlow sampling protocols. It gives those approximation statistics along with the Elasticsearch statistics. The feature bases the approximations on the flow exporter's sampling rate or a user-provided fixed factor.

Note: When the rate of flow packets is low or for short flows, the approximations will be inaccurate.

The ***Analytic Node DMF 8.5.0*** release does not support the automated approximation of total bytes and packets for Netflow v9/IPFIX. If upsampling is needed, Arista Networks recommends configuring a fixed upsampling rate.

NetFlow/IPFIX Configuration

To perform the Kibana configuration, select the **System** > **Configuration** tab on the Fabric page and open the **Analytics Configuration** > **netflow_stream** visualization.

Figure 7-10: Dashboard - Netflow IPFIX configuration

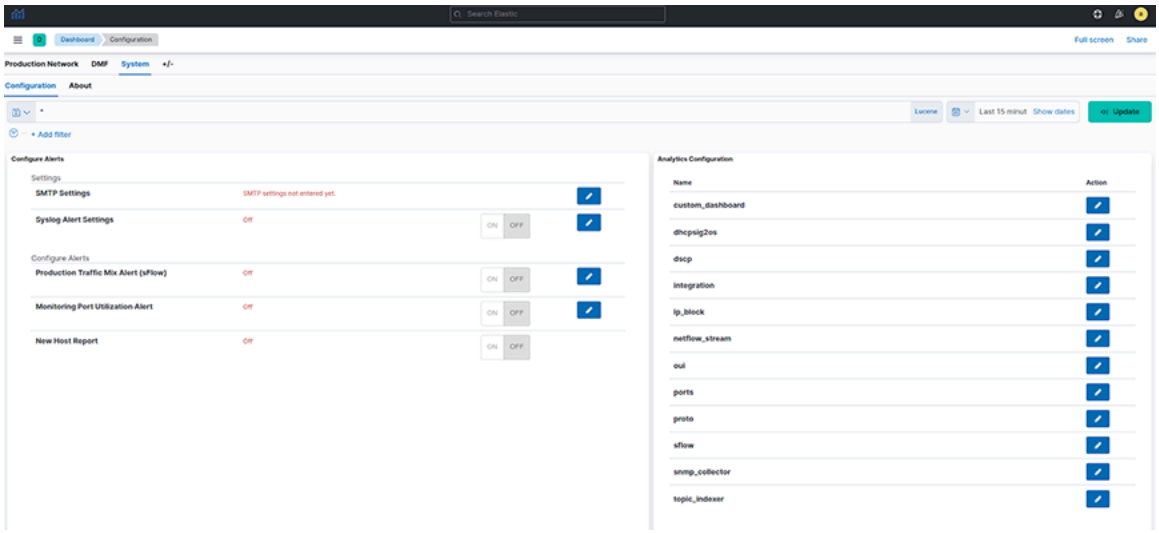
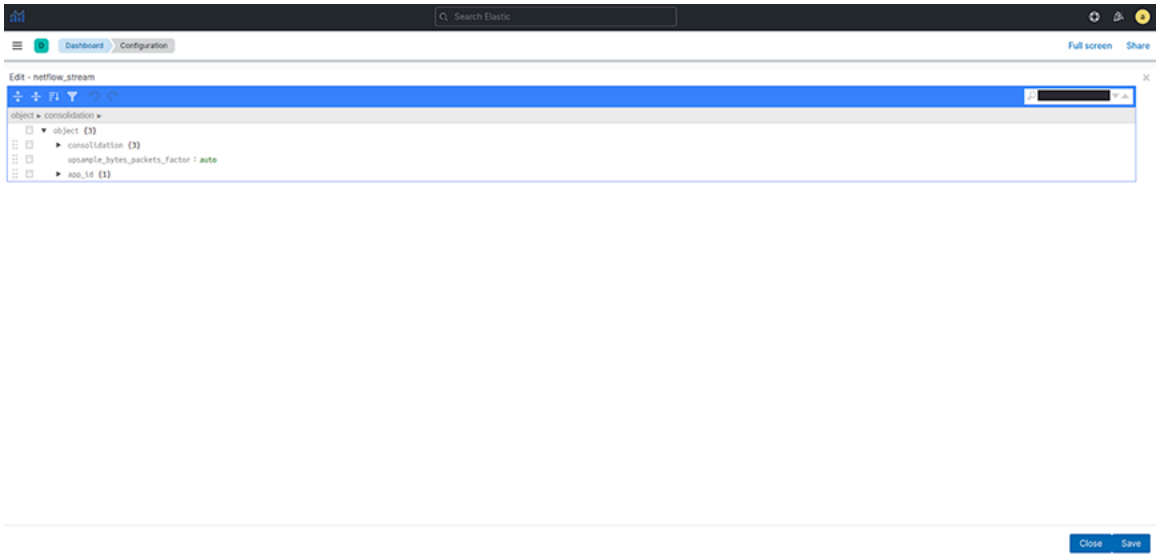


Figure 7-11: Edit - Netflow IPFIX



There is one required setting, ***upsample_byte_packet_factor***, with two possible options:

- **Auto**: This is the default option. Arista Networks recommends configuring an integer if upsampling is needed.
- **Integer**: Multiply the number of bytes and packets for each collected sample using this configured number.

sFlow Configuration

To perform the Kibana configuration, select the **System** > **Configuration** tab on the Fabric page and open the **Analytics Configuration** > **sFlow** visualization.

Figure 7-12: Dashboard - sFlow configuration

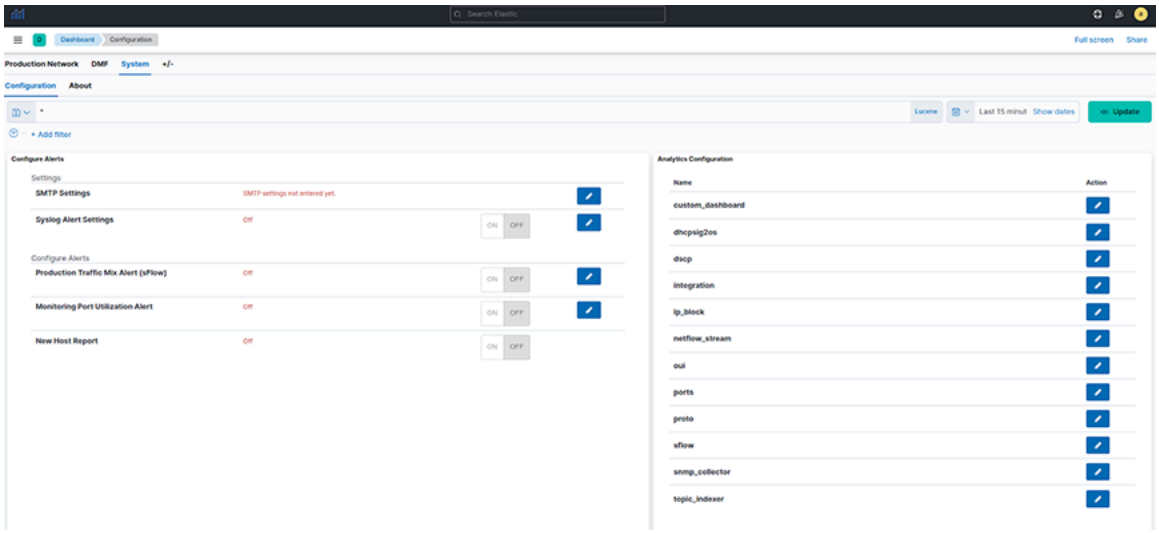
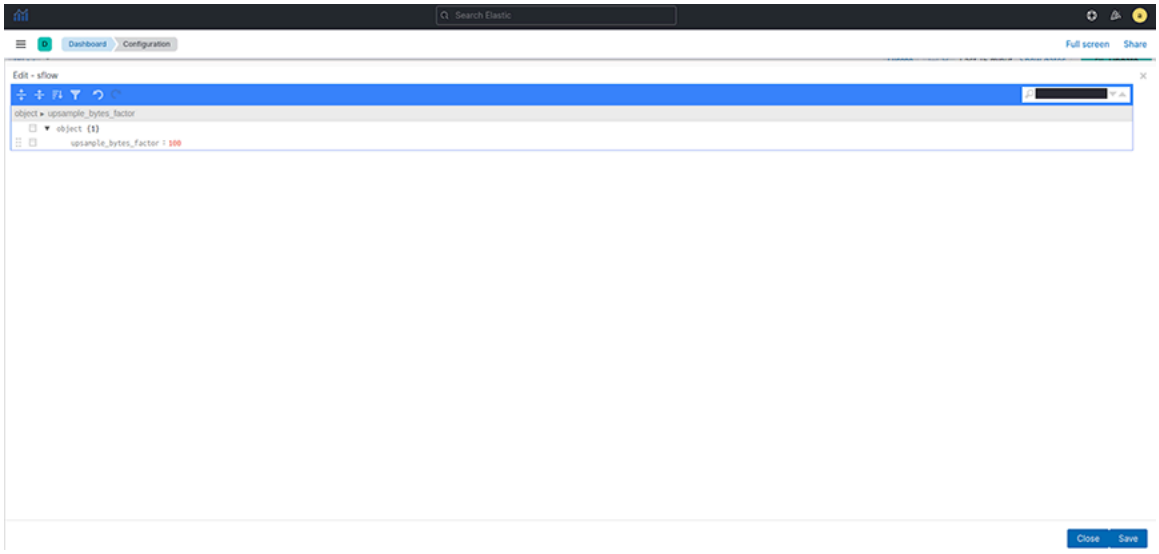


Figure 7-13: Edit - sFlow



There is one required setting, ***upsample_byte_packet_factor***, with two possible options:

- **Auto**: Approximate the bytes and packets for each collected sample based on the collector's sampling rate. **Auto** is the default option.
- **Integer**: Multiply the number of bytes and packets for each collected sample using this configured number.

Troubleshooting

Arista Networks recommends creating a support bundle and contacting Arista Networks TAC if upsampling isn't working correctly.

7.1.4 Non-standard Ports Support for IPFIX and NetFlow v5

Often, there is a need to accept IPFIX/NFv9 and NFv5 traffic arriving at ports other than the standard ports 4739 and 2055, respectively. To address this need, DMF allows the following non-standard ports to forward traffic to their standard ports on the physical IP of the Analytics Node (AN) and the cluster's Virtual IP (VIP).

- NetFlow v5 (Standard Port: 2055): Alternates 1 and 2 are UDP ports 9555 and 9025.
- NetFlow v9/IPFIX (Standard Port: 4739): Alternates 1 and 2 are UDP ports 9995 and 9026.

Configuration

The alternate ports do not require special configuration; DMF automatically configures them to allow Netflow from any subnet, as illustrated in the following **`show`** command outputs.

Show Commands

Refer to the controls for alternate ports in the following output example configuration for ***alt1*** and ***alt2***:

```
ana1ytics(config-cluster-access)# show this
! cluster
cluster
  access-control
    access-list active-directory
    !
    access-list api
      1 permit from ::/0
      2 permit from 0.0.0.0/0
    !
    access-list gui
      1 permit from ::/0
      2 permit from 0.0.0.0/0
    !
    access-list ipfix
      1 permit from ::/0
      2 permit from 0.0.0.0/0
    !
    access-list ipfix-alt1
      1 permit from ::/0
      2 permit from 0.0.0.0/0
    !
    access-list ipfix-alt2
      1 permit from ::/0
      2 permit from 0.0.0.0/0
    !
    access-list netflow
      1 permit from ::/0
      2 permit from 0.0.0.0/0
    !
    access-list netflow-alt1
      1 permit from ::/0
      2 permit from 0.0.0.0/0
    !
    access-list netflow-alt2
      1 permit from ::/0
      2 permit from 0.0.0.0/0
    access-list redis
    !
    access-list replicated-redis
    !
    access-list snmp
      1 permit from ::/0
      2 permit from 0.0.0.0/0
    !
    access-list ssh
      1 permit from ::/0
      2 permit from 0.0.0.0/0
```

7.2 Displaying Flows with Out-Discards

The **NetFlow** dashboard allows displaying flows with out-discards when the NetFlow packets come from third-party devices. To display this information, select the **flows via interfaces with SNMP out-discards** tab at the top of the Arista Analytics **NetFlow** dashboard.

To display the flows with out-discards, click the **flows via interfaces with SNMP out-discards** tab and click the **Re-enable** button. This window displays the flows with out-discards. This capability is valuable for network monitoring and troubleshooting as out-discards on interfaces can indicate potential issues such as congestion, rate limiting, errors or configuration issues.

7.3 Latency Differ and Drop Differ Dashboard

The DANZ Monitoring Fabric (DMF) Latency Differ Dashboard and Drop Differ Dashboard feature provides a near real-time visual representation of latency and drops in the DMF Analytics Node (AN) dedicated to NetFlow Records.

For a given flow, it reports the latency and drop of packets over time between two existing tap points (**A**, **B**), with network flows traversing the managed network from **A** towards **B**.

This feature introduces the concept of **DiffPair**, defined as a flow from **A** towards **B**.

The Dashboards provide comprehensive information about the flows. The data helps determine which applications are running slow and identifies peak times. A configurable mechanism alerts on abnormal drops and latency.

Introducing DiffPair

When identifying the flows between two tap points or filter interfaces, the aggregation occurs as **A** towards **B** pairs. It implies that point **B** receives a flow originating from point **A**. The term **DiffPair** is employed to visualize this flow as a cohesive set. This newly introduced field in the flow data selects the ingress and egress tap points encompassing a flow in between. The utilization of this **DiffPair** facilitates tap point filtering and comparison.

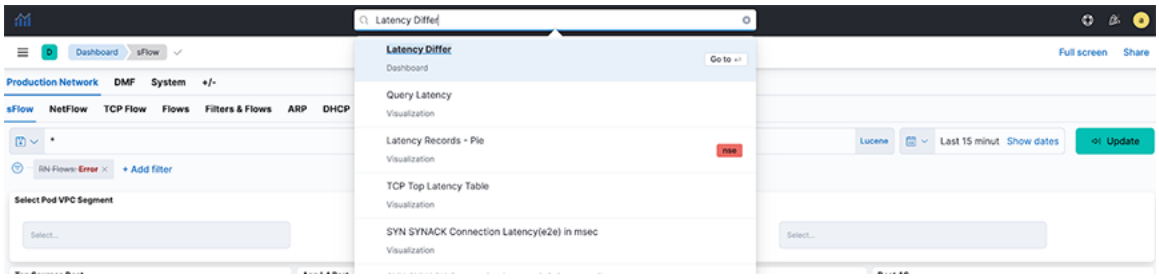
Note: It is important to verify the accuracy of the **DiffPair** data flowing between the tap points when comparing source data to the destination data.

Latency Differ Dashboard

Locate the **Latency Differ** dashboard by searching for the term **Latency Differ**.

The dashboard combines a visual representation of NetFlow Latency data in two views. The upper view displays individual flows, while the lower view aggregates A towards B pairs (**A** > **B**) or **DiffPair**.

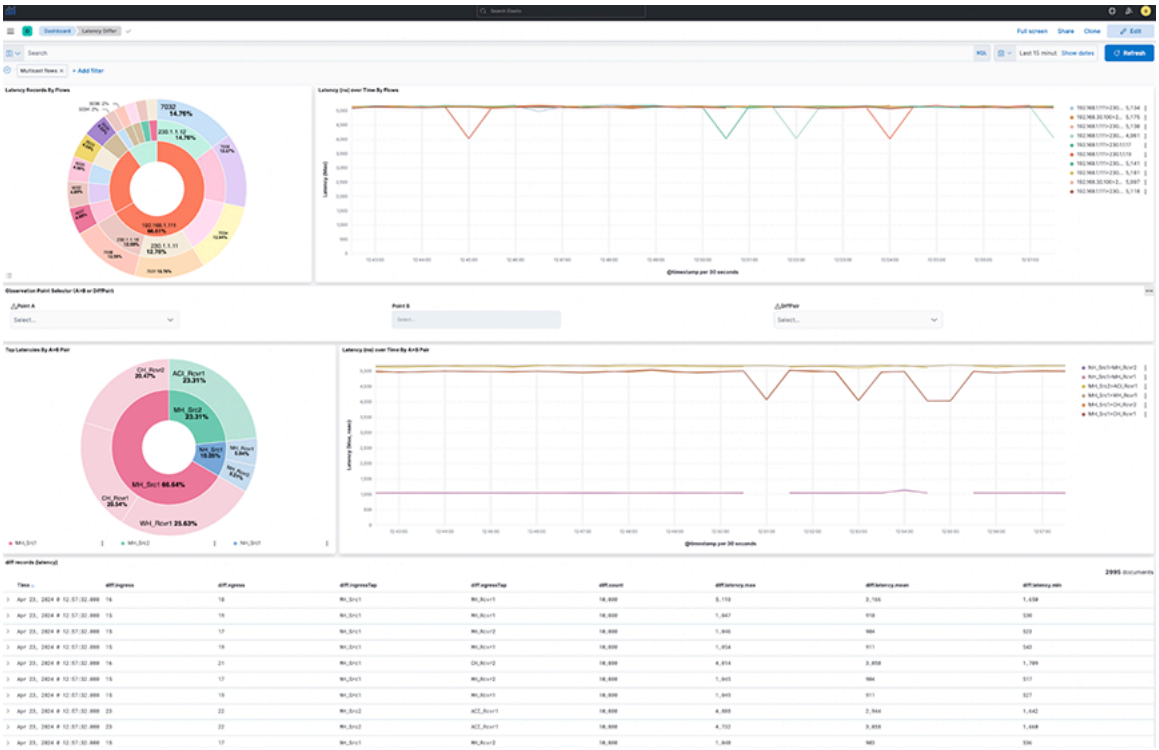
Figure 7-14: Latency Differ Dashboard



The following widgets appear in the Latency Differ dashboard:

- **Latency Records By Flows:** The pie chart represents the proportions of flow latency summed. The inner circle displays source IP addresses, the middle circle displays destination IP addresses, and the outermost circle displays destination ports.
- **Latency over time By Flows**: The line chart represents the maximum Latency in nanoseconds (ns) over time, split by each flow between source IP and destination IP addresses.
- **Observation Point Selector (A > B or DiffPair):** Use the drop-down menus to filter by **A** > **B** pair or **DiffPair**. The point **B** selector is dependent on point **A**.
- **Top Latencies By A > B Pair:** The pie chart shows the Latency max summed by **A** > **B** Points. The inner circle displays the source **A** tap point, while the outer circle displays the **B** destination tap point.
- **Latency over time By A > B Pair:** The line chart represents the maximum Latency in nanoseconds (ns) over time, split by each **A** > **B** pair between the source tap point and destination tap point.

Figure 7-15: Latency Record by Flows



Select **A** > **B** selection or **DiffPair** to visualize the data types. Filter the data using **A** > **B** Points by selecting a single source (**A**) and one or more receivers (**B**).

Figure 7-16: Flow Record with Observation Point Selector -Top

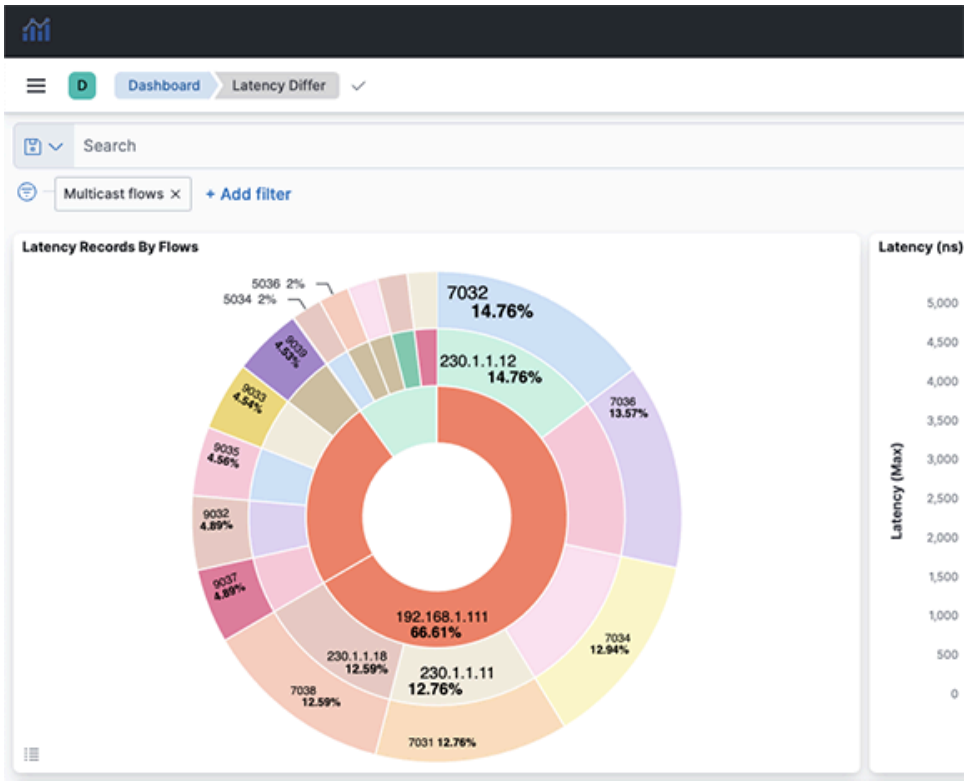


Figure 7-17: Flow Record with Observation Point Selector - Bottom

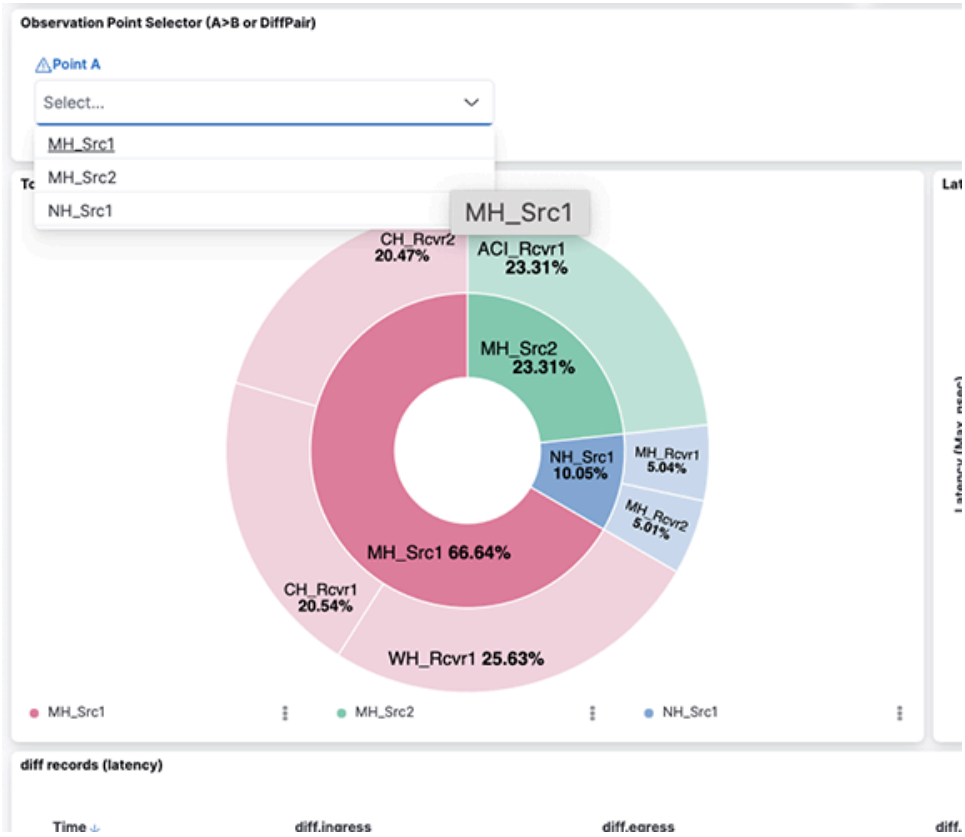
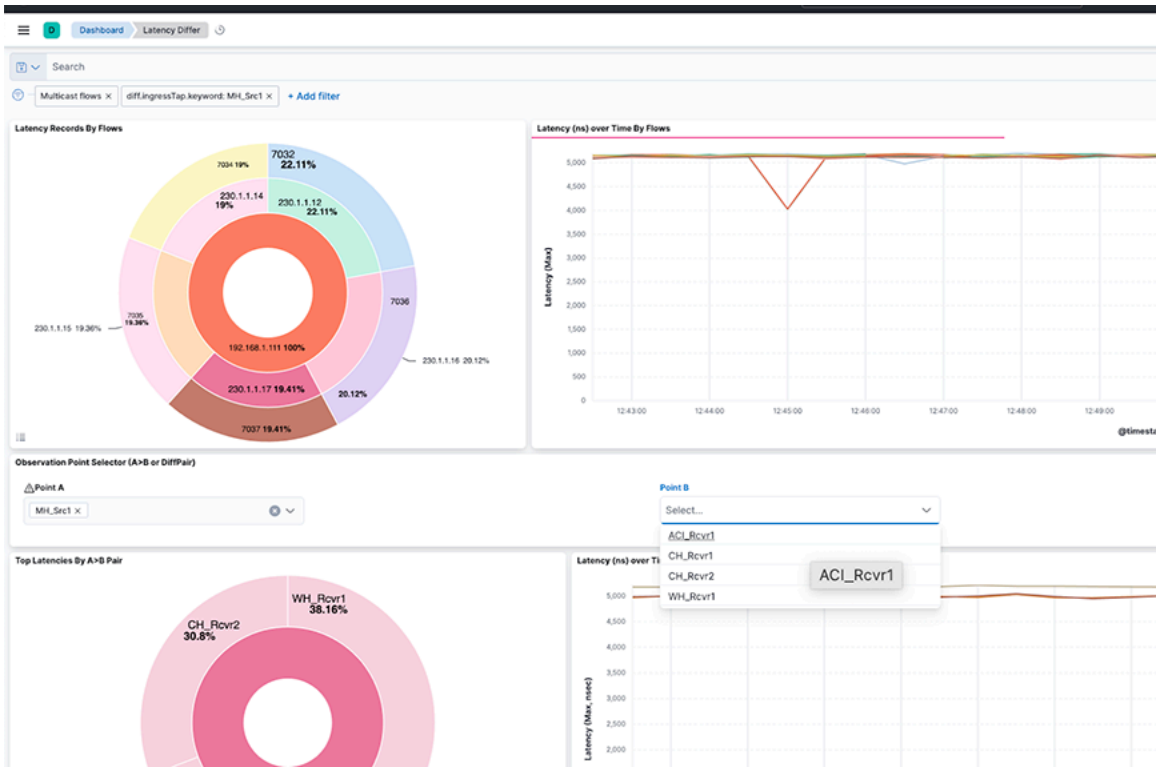
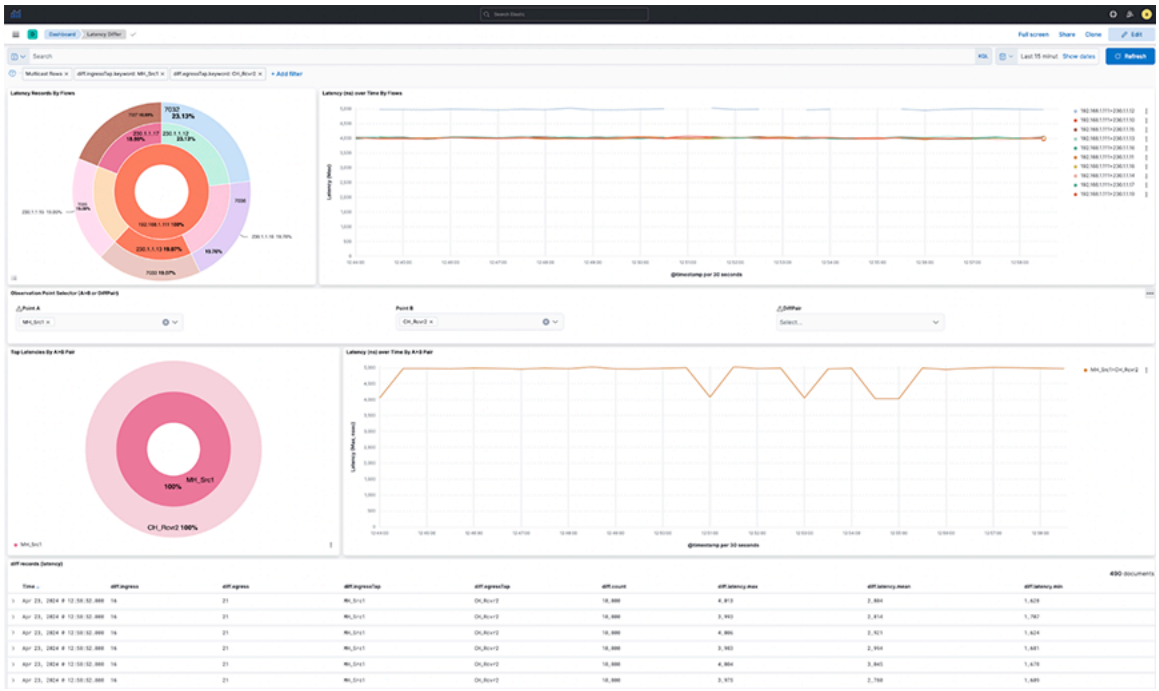


Figure 7-18: Latency between Points



The dashboard displays the latency between points **A** and **B**(s), separated by flows between the points in the upper view or filtered by the **A** > **B** pairs in the lower view. The diff records appear on the lower dashboard.

Figure 7-19: Diff Record over Time



Select individual data points in the visualization for further analysis.

Change the visualization perspective by selecting DiffPairs by selecting one or more **DiffPair** for their analysis.

Figure 7-20: DiffPair Analysis

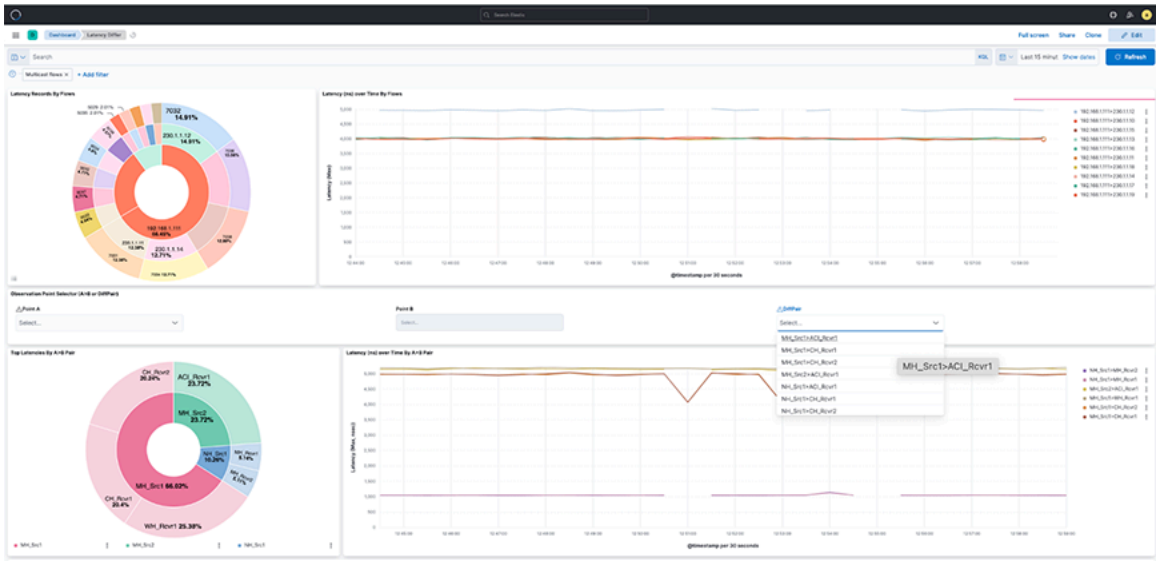
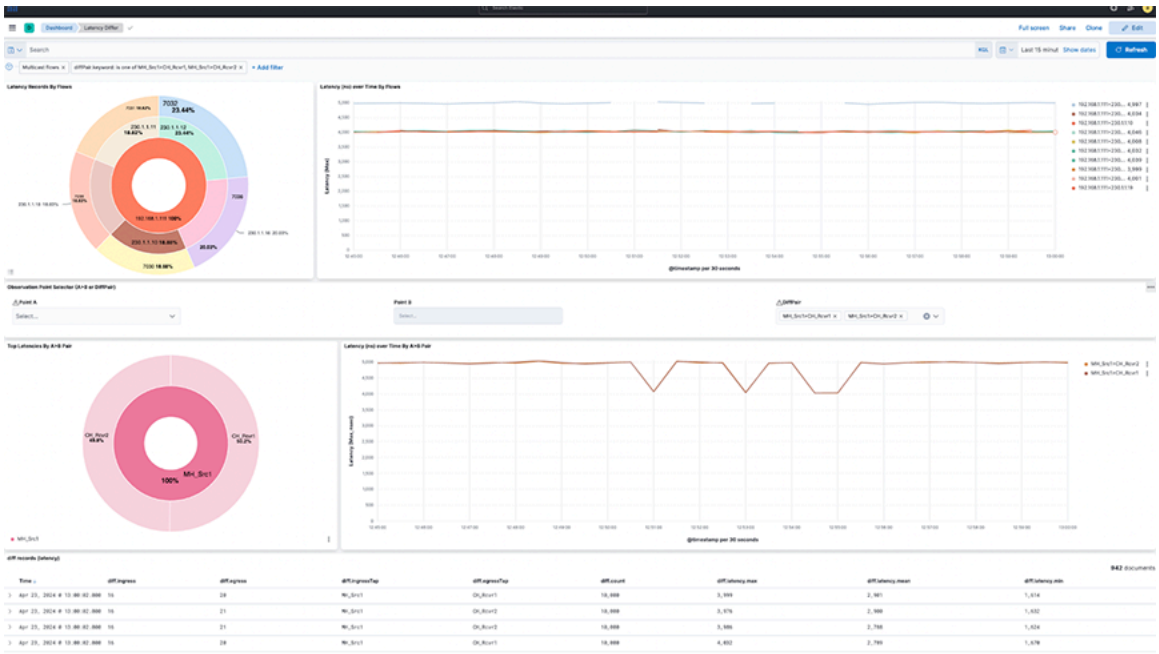


Figure 7-21: Another DiffPair Analysis

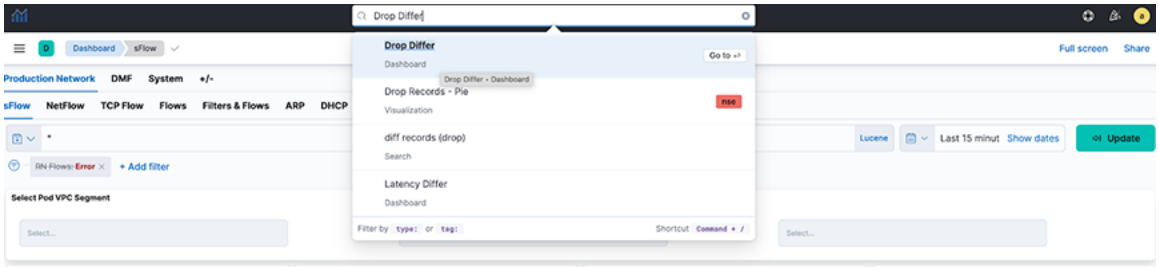


Drop Differ Dashboard

Locate the **Drop Differ** dashboard by searching for the term **Drop Differ**.

The dashboard combines a visual representation of NetFlow Latency data in two views. The upper view displays individual flows, while the lower view aggregates A towards B pairs (**A** > **B**) or **DiffPair**.Drop Differ Dashboard

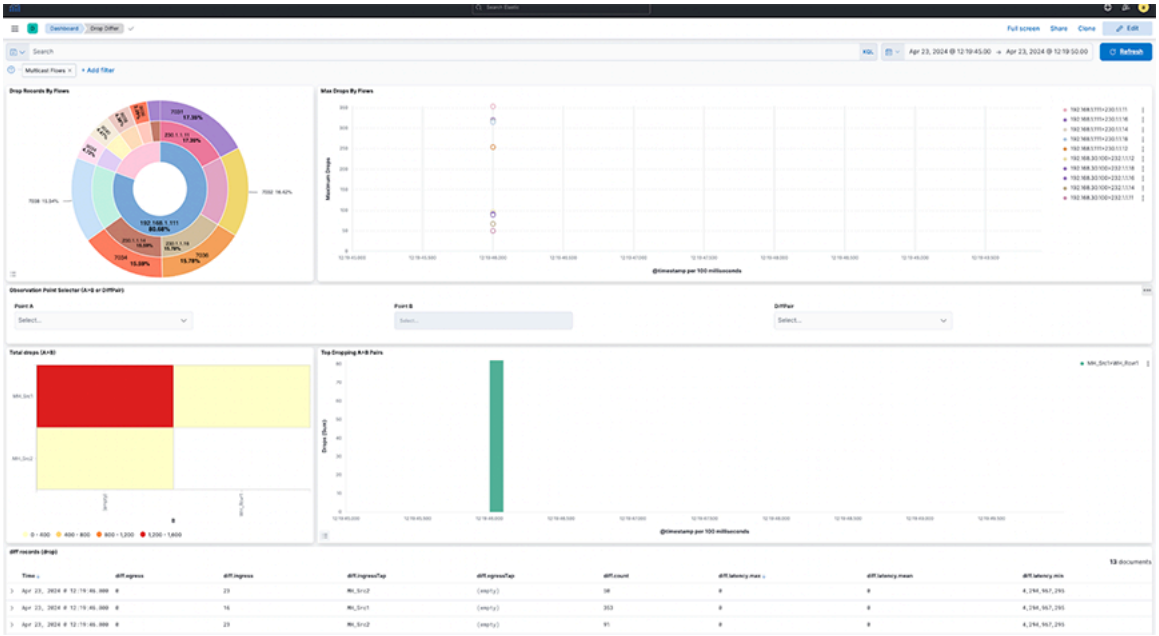
Figure 7-22: Drop Differ Dashboard



The following widgets appear in the Drop Differ dashboard:

- **Drop Records By Flows:** The pie chart represents the proportions of drop packets for each flow summed. The inner circle displays source IP addresses, the middle circle displays destination IP addresses, and the outermost circle displays destination ports.
- **Max Drops By Flows**: The line chart represents the maximum number of drop packets, separated by each flow between source IP and destination IP addresses. If fewer data points exist, the chart displays them as individual points instead of complete lines.
- **Observation Point Selector (A>B or DiffPair):** Use the drop-down menus to filter by **A** > **B** pair or **DiffPair**. The point **B** selector is dependent on point **A**.
- **Top Drop A>B:** The heat map displays the drop of packets summed by **A** > **B** Points. The map plots the source tap point, **A**, on the vertical axis and the destination tap point, **B**, on the horizontal axis.
- **Top Dropping A>B Pairs:** The bar chart represents the sum of drop packets over time, separated by each **A** > **B** pair between the source and destination. It shows the **Top 10** available dropping **A** > **B** pairs.

  Figure 7-23: Top Dropping A>B Pairs



Select **A** > **B** selection or **DiffPair** to visualize the data types.

Filter the data using **A** > **B** Points by selecting a single source (**A**) and one or more receivers (**B**).

Figure 7-24: Data Types Visualization - Top

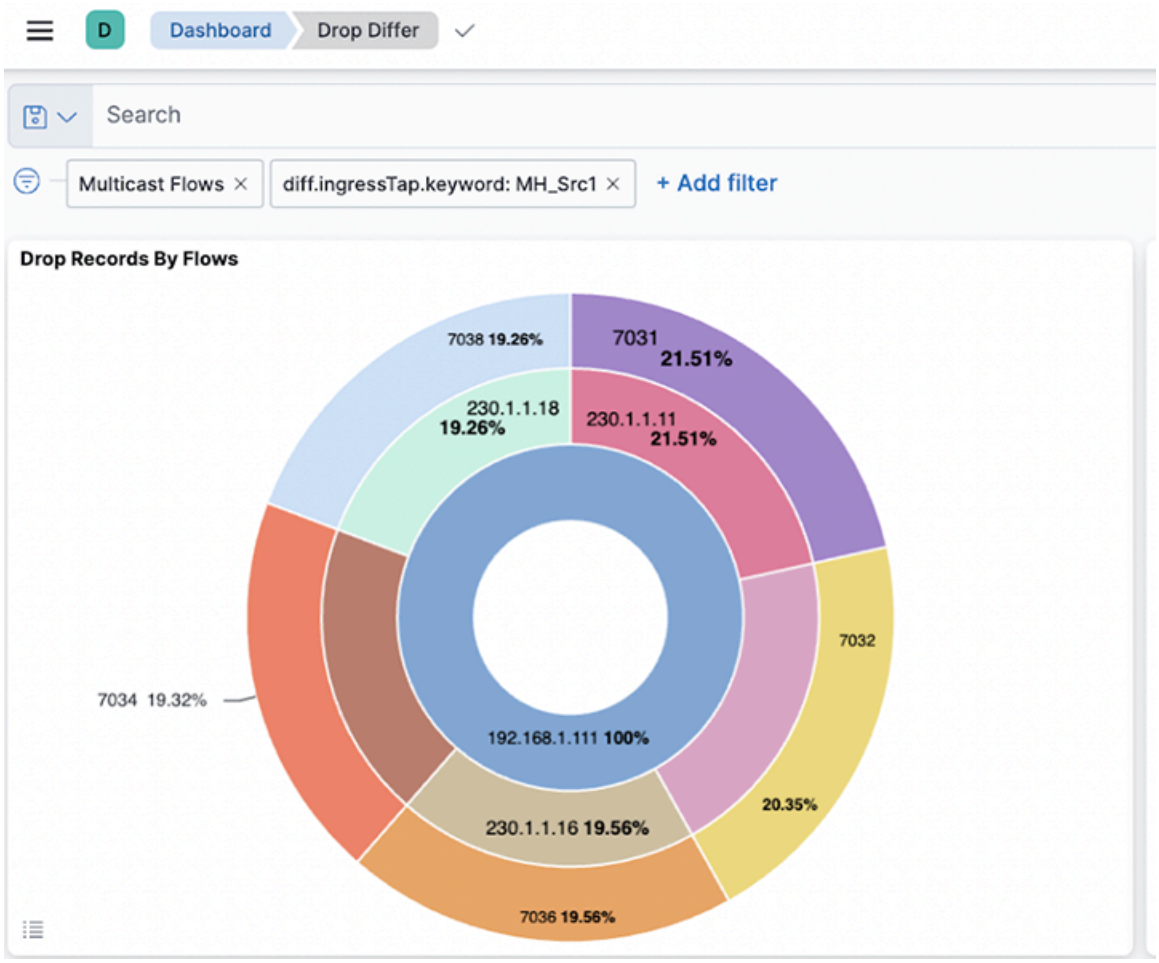
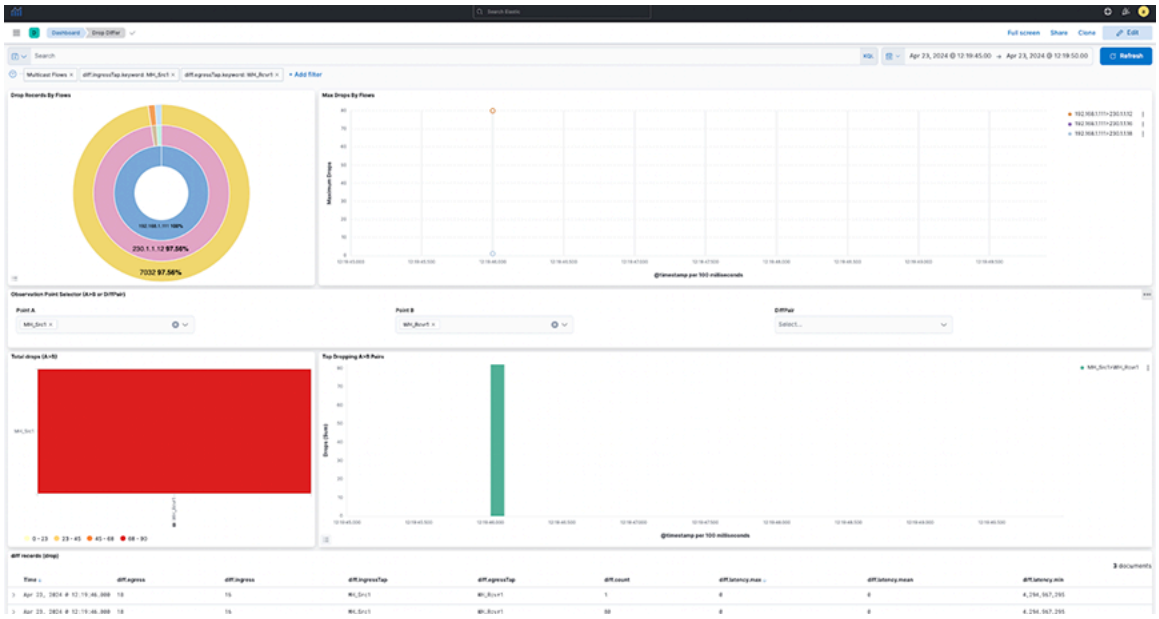


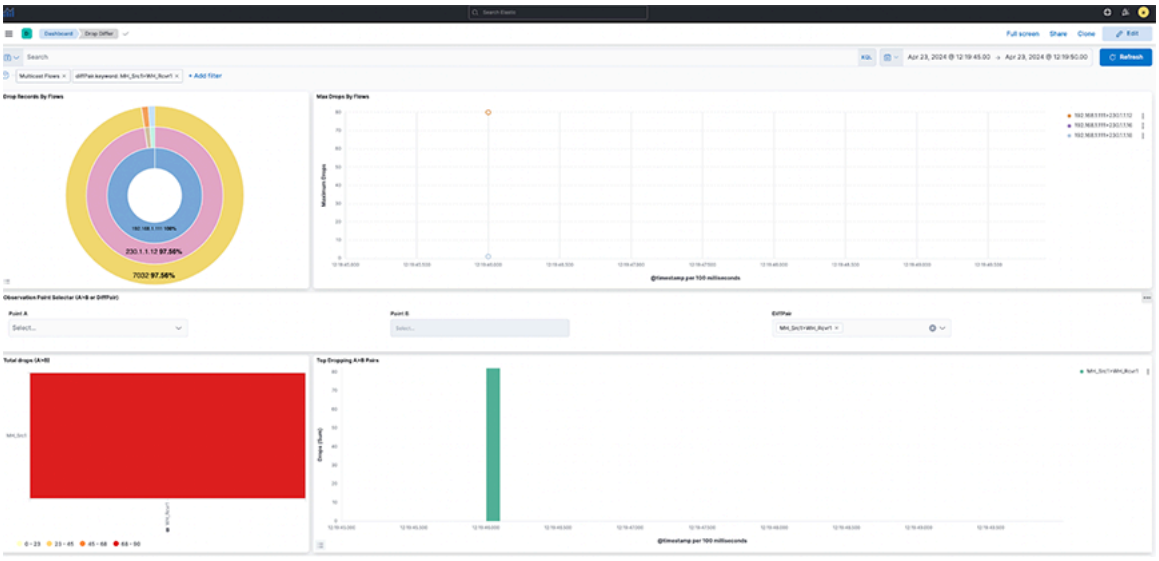
Figure 7-25: Data Types Visualization - Bottom

Figure 7-26: Single Source Data



- It provides a dashboard for packet drops between points A and B(s), either split by flows in between those points (Top) or filtered by **A** > **B** pairs (bottom) as selected. View the diff records at the bottom of the dashboard.
- Select individual data points in the visualization for further analysis.
- Selecting DiffPairs can provide a similar visualization perspective. Choose one or more DiffPairs for analysis.

Figure 7-27: DiffPair Analysis for Drop Differ



Configuring Watcher Alerts

Watcher is an elastic search feature that supports the creation of alerts based on conditions triggered at set intervals. For more information, refer to the https://www.elastic.co/guide/en/kibana/8.15/watcher-ui.html.

AN includes two built-in examples of watcher templates for ease of use. To access the templates, navigate to **Stack Management** > **Watcher**.

- Arista_NetOps_Drop_Differ_Watch
- arista_NetOps_Latency_Differ_Watch

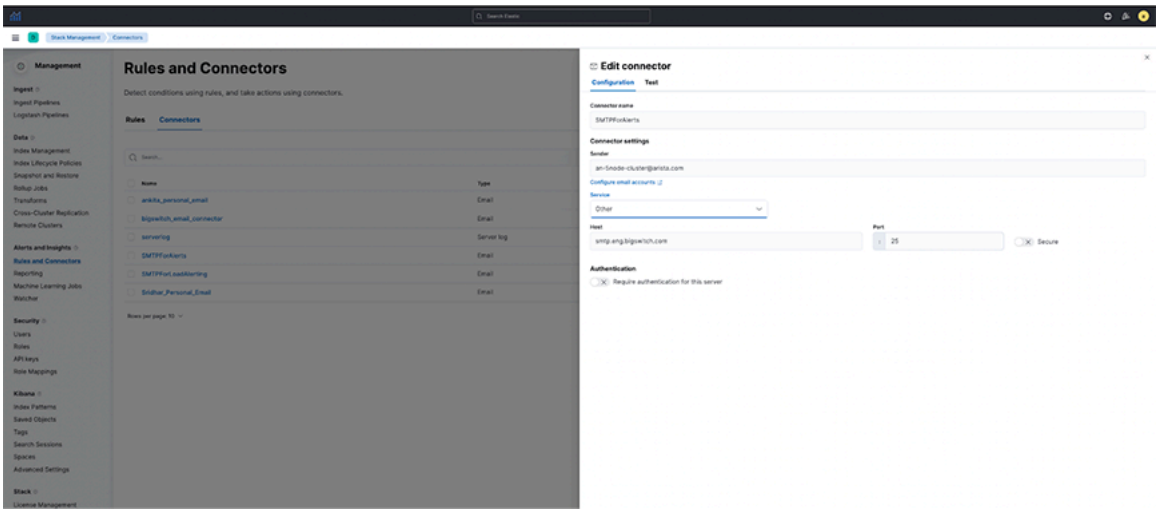
By default, it disables the templates and requires manual configuration before use.

Setting the SMTP Connector

The system dispatches **Alerts** by email; configure the **SMTPForAlerts Connector** before use.

1. Navigate to **Stack Management** > **Connector**.
2. Under **Configuration** for the **SMTPForAlerts Connector**, specify the **Sender**and **Service** field values.
3. Email alerts may require authentication based on the type of mail service selected.
4. Test and validate the settings using the **Test** tab.

Figure 7-28: Testing SMTP Connector



Setting the Watchers

- **arista_NetOps_Drop_Differ_Watch**:
  1. It configures the watcher to send an alert when the maximum drop count of packets in **NetFlow** in the last 5-minute interval exceeds the historical average (last 7-day average) of drop of packets by a ***threshold percentage***.
  2. By default, it configures the watcher to be **triggered** every 10 minutes.
  3. As this may be incorrect for all flows combined, configure it for a particular **Flow** and **Destination Port**.
  4. Search for **CHANGE_ME** in the watcher and specify the flow and destination port value (introduced to correctly compare each flow and destination port individually instead of comparing all flows together).
  5. Specify the percentage_increase parameter in the condition using a positive value between **0-100**.
  6. Enter the recipient's email address receiving the alert.

7. Select **Save watch**.

Figure 7-29: NetOps_Drop_Differ_Watch-1

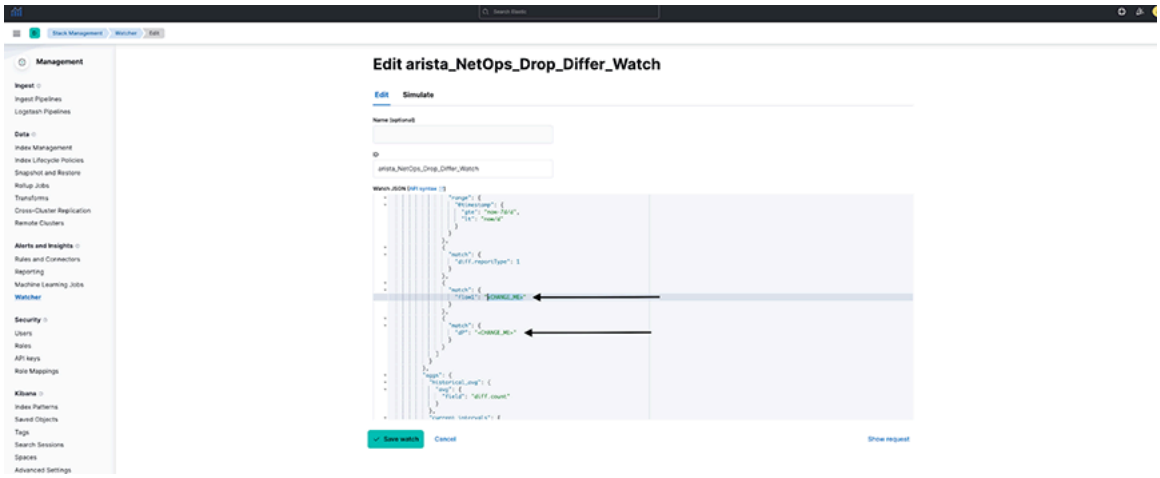


Figure 7-30: NetOps_Drop_Differ_Watch-2

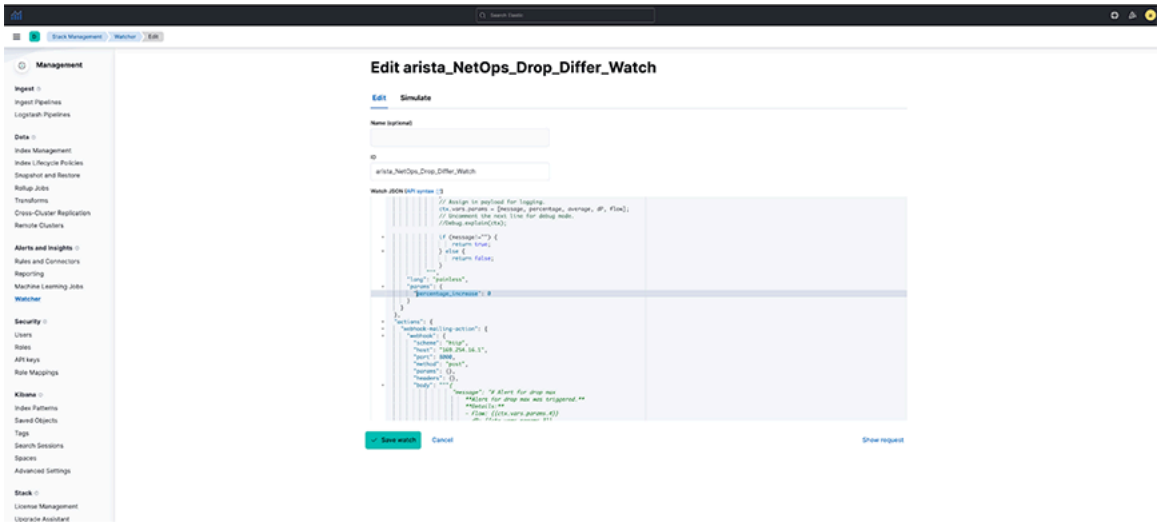
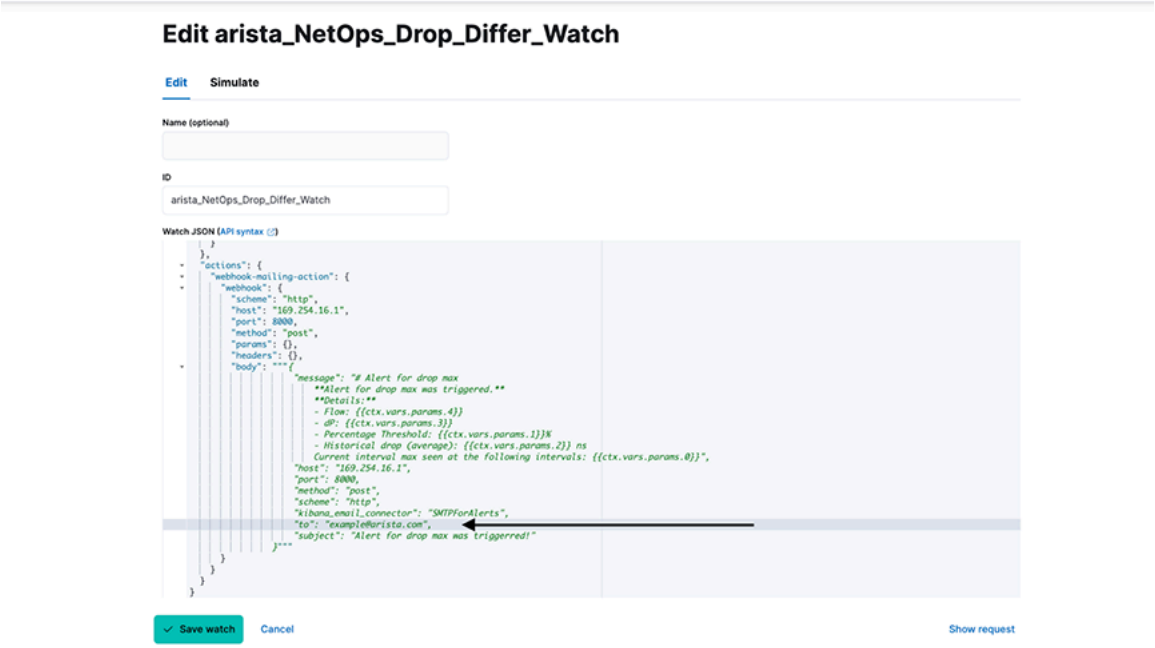


Figure 7-31: Editing NetOps_Drop_Differ_Watch

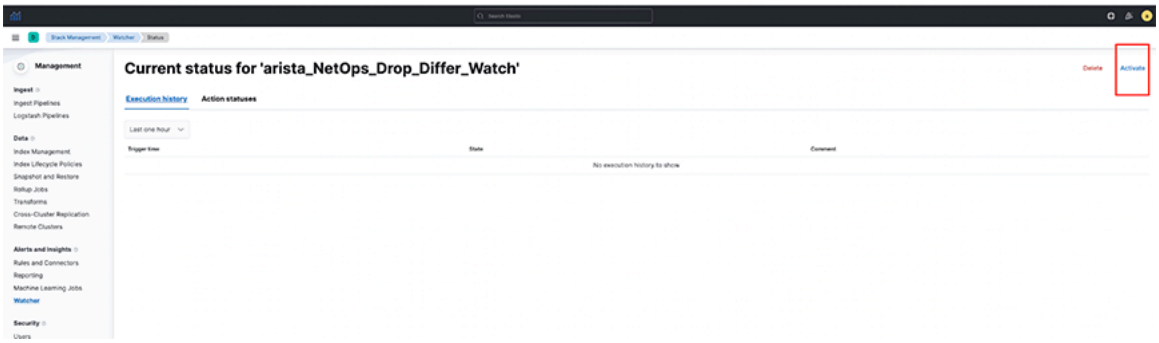


- **arista_NetOps_Latency_Differ_Watch:**
  1. It configures the watcher to send an alert when **NetFlow**'s maximum latency (or lag) in the last 5-minute interval exceeds the historical average (last 7-day average) latency by a ***threshold percentage***.
  2. By default, it configures the watcher to be **triggered** every 10 minutes.
  3. As this may be incorrect for all flows combined, configure it for a particular **Flow** and **Destination Port**.
  4. Search for **CHANGE_ME** in the watcher and specify the flow and destination port value (introduced to correctly compare each flow and destination port individually instead of comparing all flows together).
  5. Specify the percentage_increase parameter in the condition using a positive value between **0-100**.
  6. Enter the recipient's email address receiving the alert.
  7. Select **Save watch**.

Considerations

- Default Watchers are disabled and modified with user-configured alert settings before being enabled.

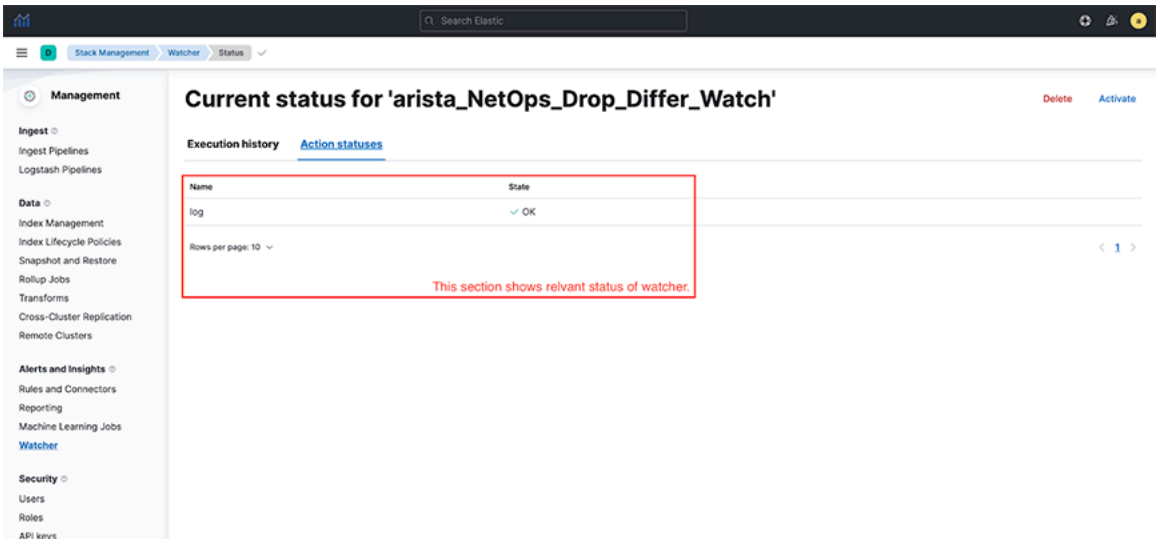
Figure 7-32: Arista_NetOps_Drop_Differ_Watch



Troubleshooting

- The dashboard obtains its data from the **flow-netflow** index. If no data is present in the dashboard, verify there is sufficient relevant data in the index.
- Watchers trigger at a set interval. To troubleshoot issues related to watchers, navigate to **Stack Management** > **Watcher**. Select the requisite watcher and navigate to **Action statuses** to determine if there is an issue with the last trigger.

Figure 7-33: Watcher Action Status



Usage Notes

- The dashboards only show partial and not full drops during a given time and are configured with filtering set to the **egress.Tap** value as **empty**.
- A **full drop** occurs when the flow of packets is at the source tap point, but no packet is at the destination tap point. The dashboards are configured to filter out full drop flows.
- A **partial drop** is a scenario in which the flow of packets at the source tap point and some, if not all, packets are observed at the destination tap point. The dashboards clearly show partial drop flows.

Chapter 8

Monitoring Active Directory Users

Windows Active Directory should be configured to audit logon and logoff events on Active Directory.

1. Download and install Winlogbeat from the Elastic website on the Windows machine. Download Winlogbeat.
2. On the Analytics node, run: **`sudo rm -rf *`** inside **`/home/admin/xcollector`** and then run **`docker exec xcollect /home/logstash/generate_client_keys.sh <AN IP> client`**. It generates **.pem** files in **`/home/admin/xcollector`**.
3. On the Analytics node machine, replace the **`winlogbeat.yml`** file from **`/opt/bigswitch/conf/x_collector/winlogbeat.yml`** to the one in the Windows server. Edit the **logstash** output section:

```
#------------------------------ Logstash output -------------------------------
output.logstash:
#Point agent to analytics IPv4 in hosts below hosts: ["10.2.5.10:5043"]

#List of root certificates for HTTPS server verifications ssl.certificate_authorities: ["C:/Program Files/
Winlogbeat/security/ca/cacert.pem"]

#Certificate for SSL client authentication
ssl.certificate: "C:/Program Files/Winlogbeat/security/clientcert.pem"


#Client Certificate Key
ssl.key: "C:/Program Files/Winlogbeat/security/clientkey.pem"
```

4. Using the recovery account, use an SCP application to transfer the **.pem** files from the Analytics node to the Windows machine and update their locations in **`winlogbeat.yml`**.
5. On Windows, enter the powershell, navigate to **`winlogbeat.exe`**, and run: **`.install-service-winlogbeat.ps1`** to install **Winlogbeat**.
6. Test the configuration using **`"winlogbeat test config"`** to test **`winlogbeat.yml`** syntax and **`"winlogbeat test output"`** to test connectivity with **logstash** on the Analytics node.
7. Run **`winlogbeat run -e`** to start **Winlogbeat**.

Chapter 9

Machine Learning and Anomaly Detection

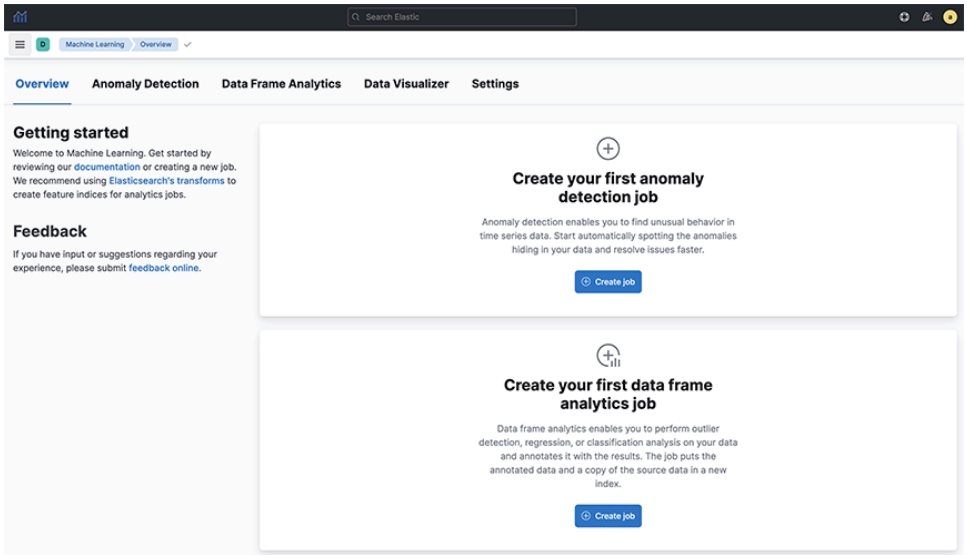
This chapter monitors network performance and identifies unusual events. It includes the following sections.

9.1 Machine Learning

Arista Analytics uses machine learning for anomaly detection. The following jobs are available:

- Single-metric anomaly detection
- Multimetric anomaly detection
- Population
- Advanced
- Categorization

Figure 9-1: Machine Learning

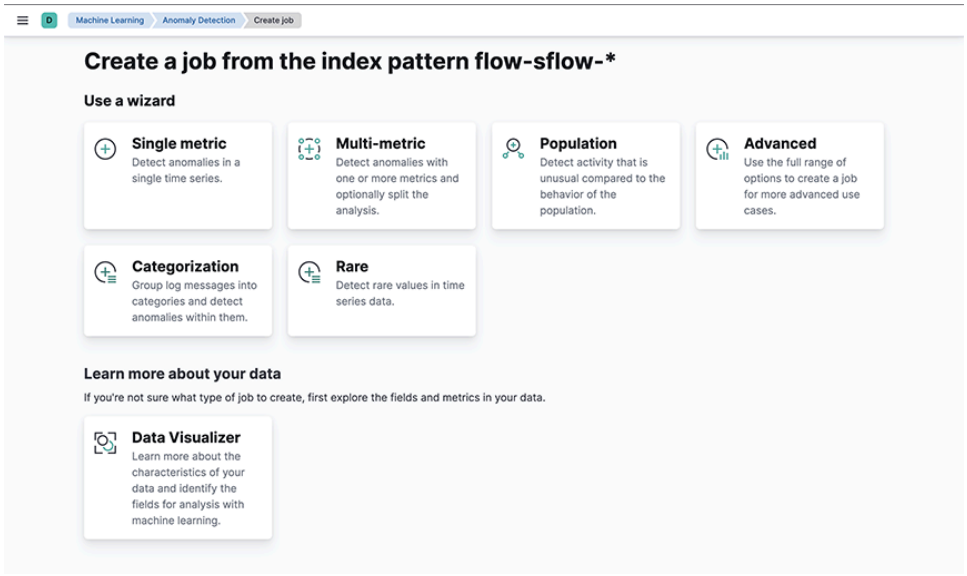


For every job, a job **ID** must be configured. To create a machine learning job:

- Select the time range
- Select the appropriate metric

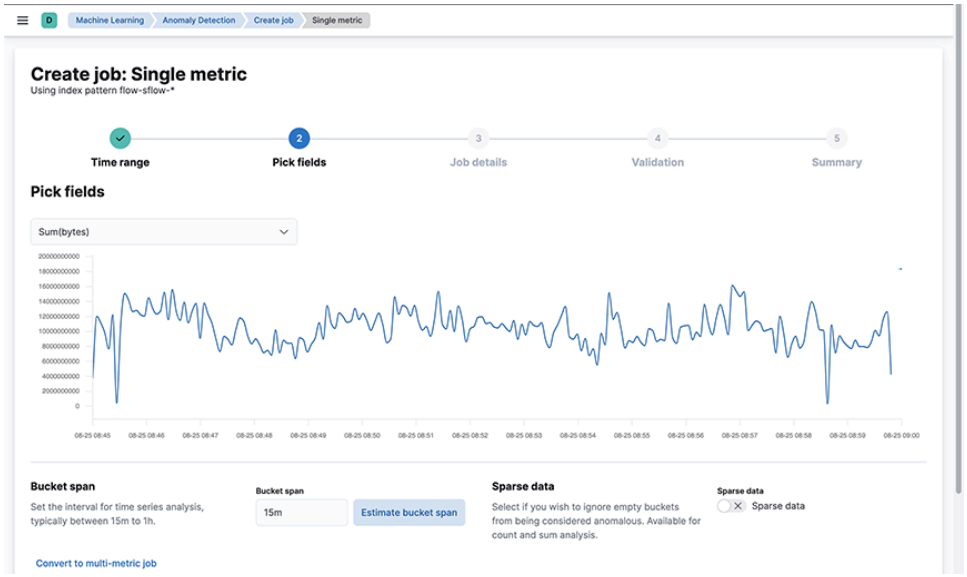
- Enter details: job ID, description, custom URLs, and calendars to exclude planned outages from the job

Figure 9-2: Machine Learning Job options



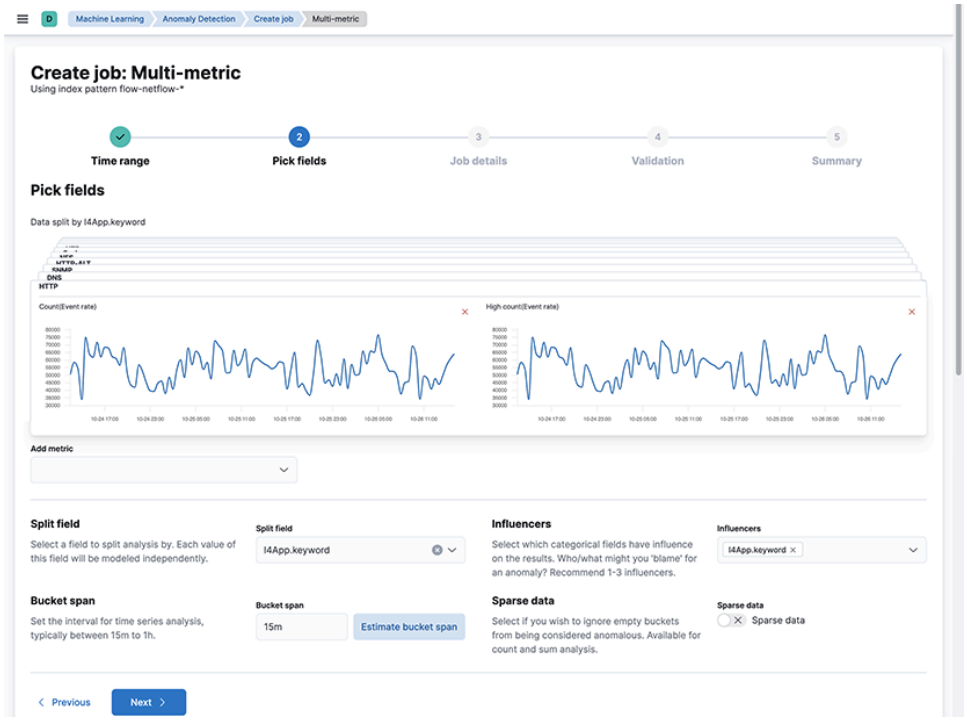
Single-metric anomaly detection uses machine learning on only one metric or field.

Figure 9-3: Single-metric Anomaly Detection



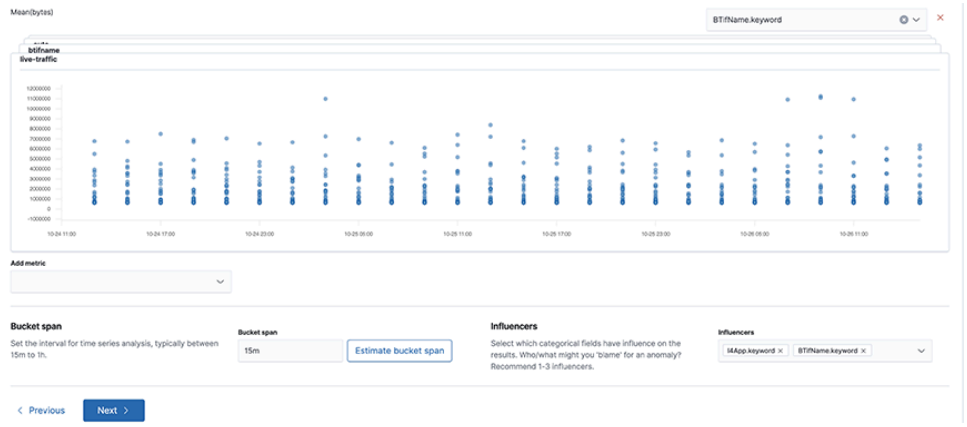
The following image uses two metrics: over and running ml per L4 app.

Figure 9-4: Multimetric Anomaly Detection



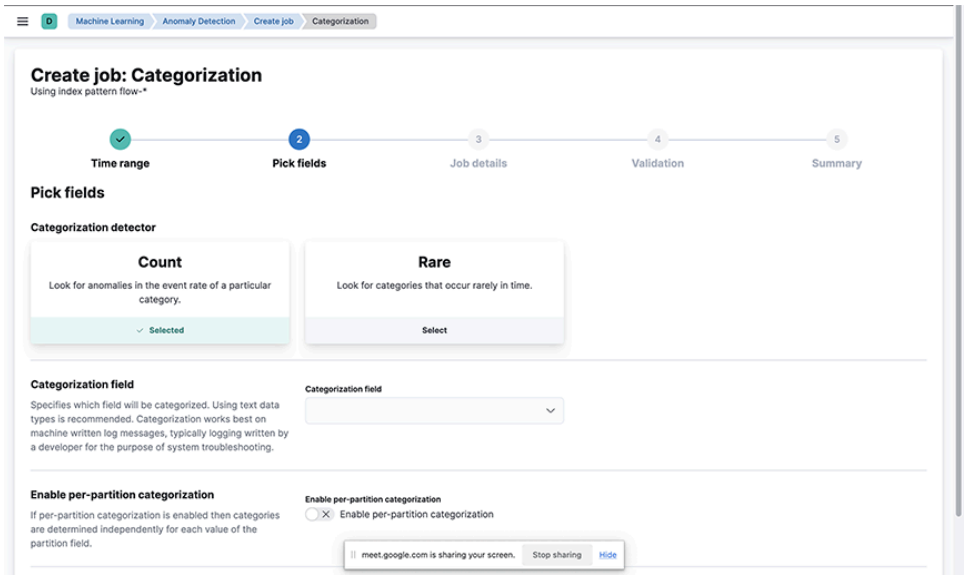
Multimetric Anomaly Detection detects network activity that differs from the population of data points. Arista Networks recommends this analysis for high-cardinality data.

Figure 9-5: Population



This job groups data points into categories and then finds anomalies between them.

Figure 9-6: Categorization



9.2 Anomalies

Use the following features to recognize unusual activity or events on the network.

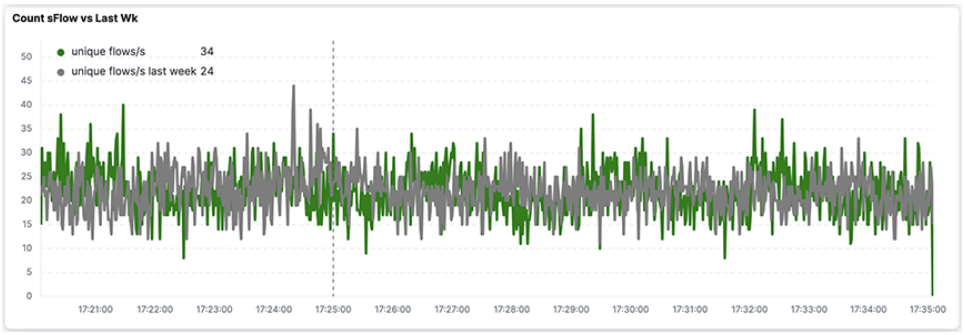
- Comparing dashboards and visualization over time
- sFlow®* > Count sFlow vs Last Wk
- New Flows & New Hosts
- Utilization alerts
- Machine Learning

Identify any unusual activity by comparing the same dashboard over the past 1 hour to the same time last week's data. For example, the bar visualization of traffic over time shows changing ratios of internal to external traffic, which can highlight an abnormality.

* sFlow® is a registered trademark of Inmon Corp.

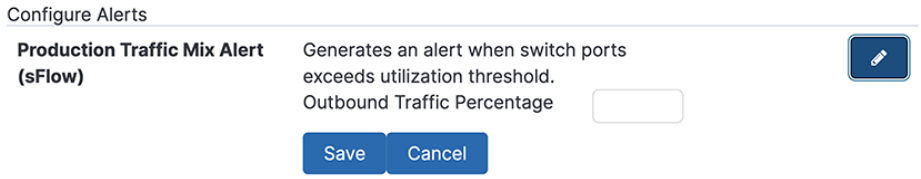
The **Count sFlow vs Last Wk** visualization in the **sFlow®** dashboard shows the number of unique flows being seen now compared to last week. This visualization indicates unusual network activity and will help pinpoint a Denial of Service (DOS) attack.

Figure 9-7: Count sFlow vs Last Wk



In a well-inventoried environment, use the **New Flows & New Hosts** report.

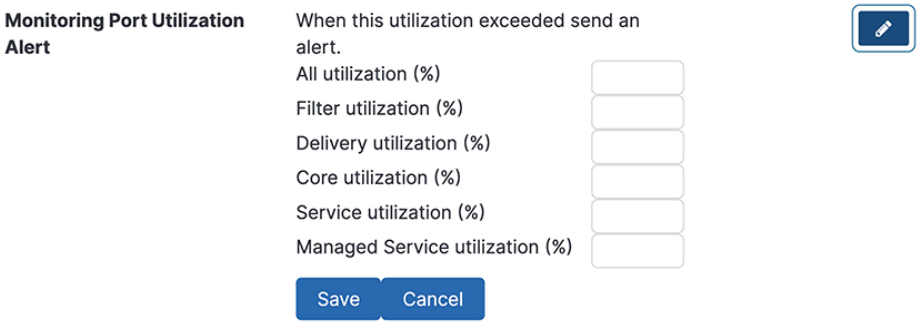
Figure 9-8: Production Traffic



Configure utilization alerts associated with the following DMF port types:

- Filter
- Delivery
- Core
- Services

Figure 9-9: Monitoring Port Utilization Alerts



The other alerts available include the following.

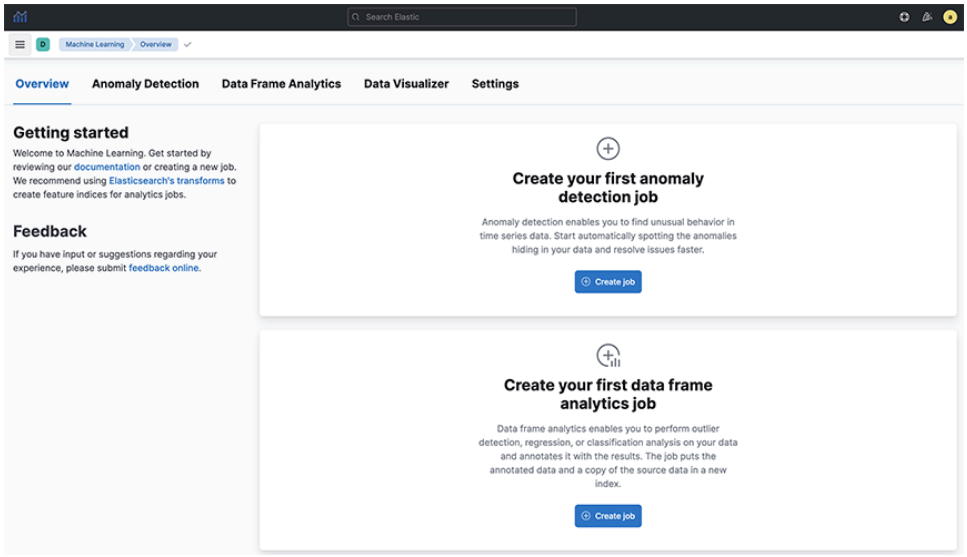
- The percentage of outbound traffic exceeds the usual thresholds.

- New hosts appear on the network every 24 hours.

Figure 9-10: New Host Report

Perform Anomaly Detection in data over byte volume and characteristics over time using machine learning.

Figure 9-11: Machine Learning



9.3 Application Data Management

Application Data Management (ADM) helps users govern and manage data in business applications like SAP ERP. To use Arista Analytics for ADM, perform the following steps:

1. Pick a service IP address or block of IP addresses.
2. Identify the main body of expected communication with adjacent application servers.
3. Filter down to ports that need to be communicating.
4. Expand the time horizon to characterize necessary communication completely.
5. Save as CSV.
6. Convert the CSV to ACL rules to enforce in the network.

Chapter 10

Backup and Restore

This chapter includes the following sections.

10.1 Elasticsearch Snapshot and Restore

Elasticsearch provides a mechanism to snapshot data to a network-attached storage device and to restore from it.

1. Mount the Network File Storage (NFS) on the Analytics Node.
   a. Create a directory on the remote Ubuntu Server (NFS store). This directory must have the user group **remoteuser** and **root**, respectively, with ***10000*** for the UID and ***0*** for the GID.
   b. Stop the Elasticsearch container: **`sudo docker elasticsearch stop`**
   c. Mount the remote store on **`/opt/bigswitch/snapshot`** in the Analytics server.
   d. Start the Analytics Node: **`sudo docker elasticsearch start`**
2. Create a snapshot repository by running the following API call:

```
curl \
-k \
-X PUT \
-H 'Content-Type:application/json' \
-d '{"type":"fs","settings":{"location":"/usr/share/elasticsearch/snapshot"}}' \
-u admin:***** \
https://169.254.16.2:9201/_snapshot/test_automation
```

3. Take a snapshot by running the following API call:

```
curl \
-k \
-X POST \
-H 'Content-Type:application/json' \
-d '{ "indices": ".ds-flow-sflow-stream-2023.08.21-000001", "include_global_state": true,
 "ignore_unavailable": true, "include_hidden": true}' \
-u admin:***** \
https://169.254.16.2:9201/_snapshot/test_automation/test_snap1
```

4. To view the a snapshot, run the following API call:

```
curl \
-s -k \
-H 'Content-Type:application/json' \
-u admin:***** \
https://169.254.16.2:9201/_snapshot/test_automation/test_snap1?pretty
```

5. To restore a snapshot, run the following API call:

```
curl \
-k \
-X POST \
-H 'Content-Type:application/json' \
-d '{ "indices": ".ds-flow-sflow-stream-2023.08.21-000001", "ignore_unavailable": true,
 "include_global_state": true, "rename_pattern": "(.+)", "rename_replacement": "restored_$1" }'
 \
-u admin:***** \
https://169.254.16.2:9201/_snapshot/test_automation/test_snap1/_restore
```

10.2 Import and Export of Saved Objects

The **Saved Objects** UI helps keep track of and manage saved objects. These objects store data for later use, including dashboards, visualization, searches, and more. This section explains the procedures for backing up and restoring saved objects in Arista Analytics.

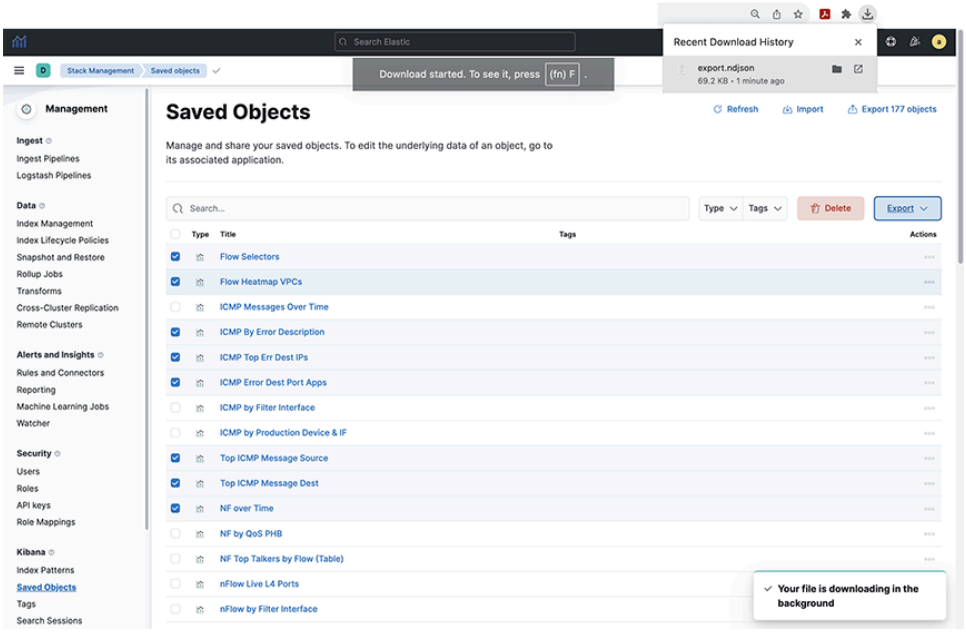
10.2.1 Exporting Saved Objects

1. Open the main menu, then click **Main Menu** > **Management** > **Saved Objects**.
2. Select the custom-saved objects to export by clicking on their checkboxes.
3. Click the **Export** button to download. Arista Networks suggests changing the file name to the nomenclature that suits your environment (for example, **`clustername_date_saved_objects_<specific_name_or_group_name>.ndjson`**).

> **Note:** Arista Networks recommends switching on to **include related objects** before selecting the **export** button. If there are any missing dependency objects, selecting **include related objects** may throw errors, in which case switch it OFF.

4. The system displays the following notification if the download is successful.

Figure 10-1: Verifying a Saved/Downloaded Object



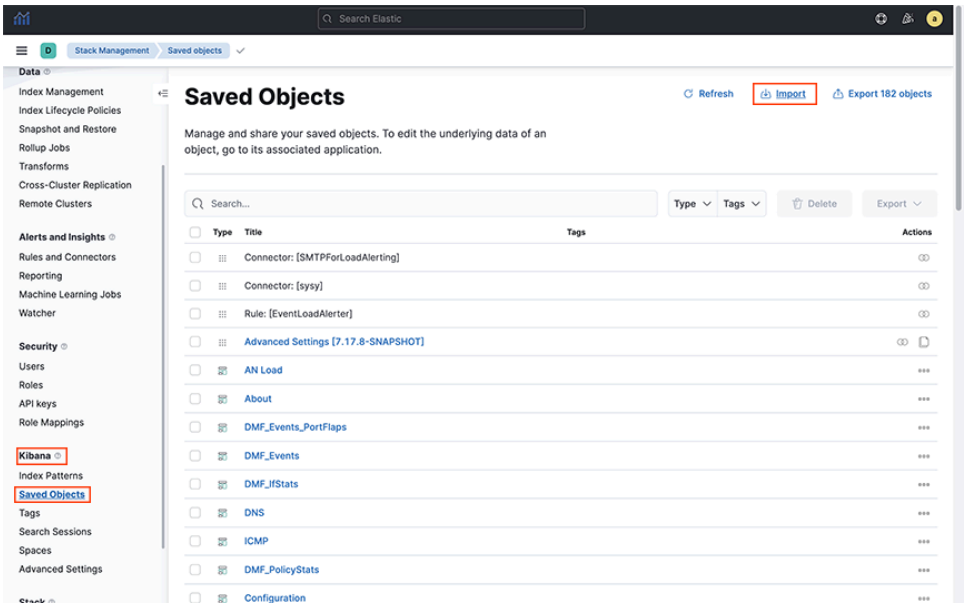
Note: Recommended Best Practices

- While creating saved objects, Arista Networks recommends naming conventions that suit your environment. For instance, in the example earlier, a naming pattern has been used, prefixed with “ARISTA” and specifying **Type: dashboard**, which allows a manageable set of items to click individually or to select all. Furthermore, exporting individual dashboards based on their **Type** is a more appropriate option, as tracking modifications to a dashboard improves using this method. Dashboards should select only custom visualizations and searches (i.e., do not depend on default objects that might change during a software upgrade).
- Do not edit any default objects. Arista Networks suggests saving the new version with a different (custom) name if default objects require editing.
- The files exported should be treated as code and reserved in a source control system, so dissimilarities and rollbacks are possible under standard DevOps approaches.

10.2.2 Importing Saved Objects

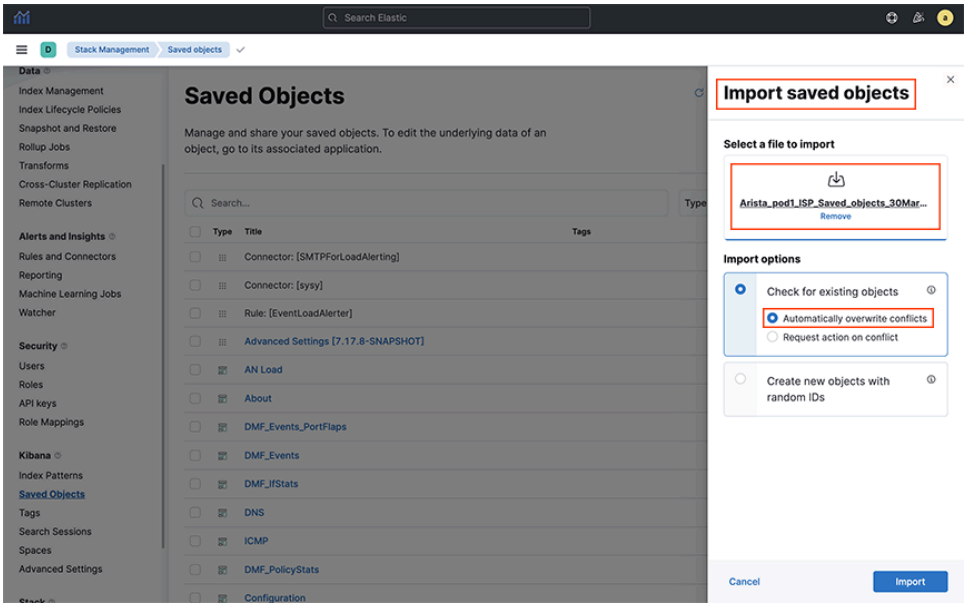
1. To import one or a group of custom-created objects, navigate to **Main Menu** > **Management** > **Kibana** > **Saved Objects**.

Figure 10-2: Importing a Group of Saved Objects



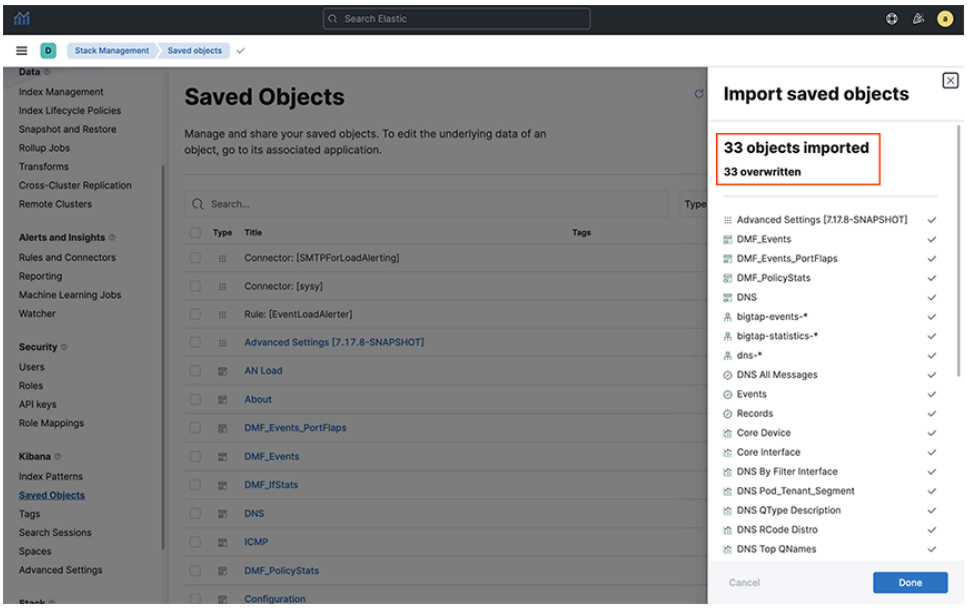
2. Click **Import** and navigate to the NDJSON file that represents the objects to import. By default, saved objects already in Kibana are overwritten by the imported object. The system should display the following screen.

Figure 10-3: NDJSON File Import Mechanism



3. Verify the number of successfully imported objects. Also verify the list of objects, selecting **Main Menu** > **Management** > **Kibana** > **Saved Objects** > **search for imported objects**.

Figure 10-4: Import Successful Dialog Box



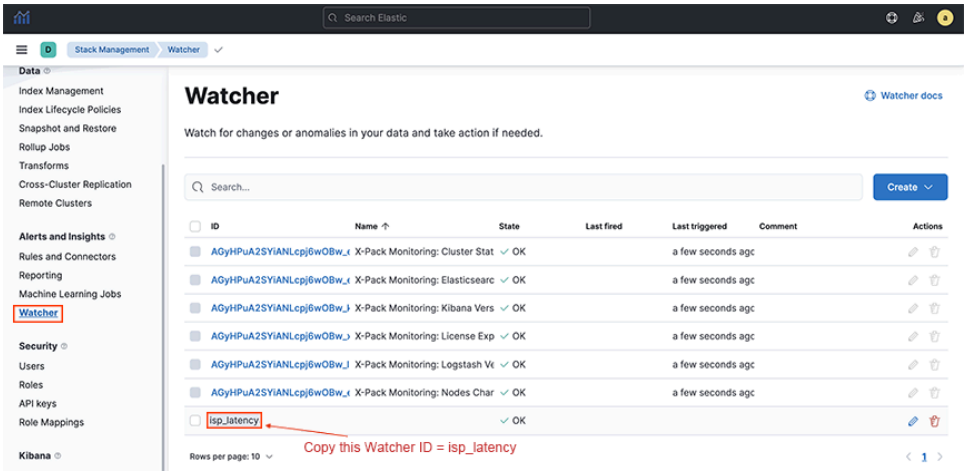
10.3 Import and Export of Watchers

Select the Watcher feature to create actions and alerts based on certain conditions and periodically evaluate them using queries on the data. This section explains how to back up and restore the Watchers in Arista Analytics.

10.3.1 Exporting Watchers

1. The path parameter required to back up the Watcher configuration is **`watcher_id`**. To obtain the **`watcher_id`**, go to **Main Menu** > **Management** > **Watcher** > **Watcher_ID**.

   Figure 10-5: Find Watcher_ID



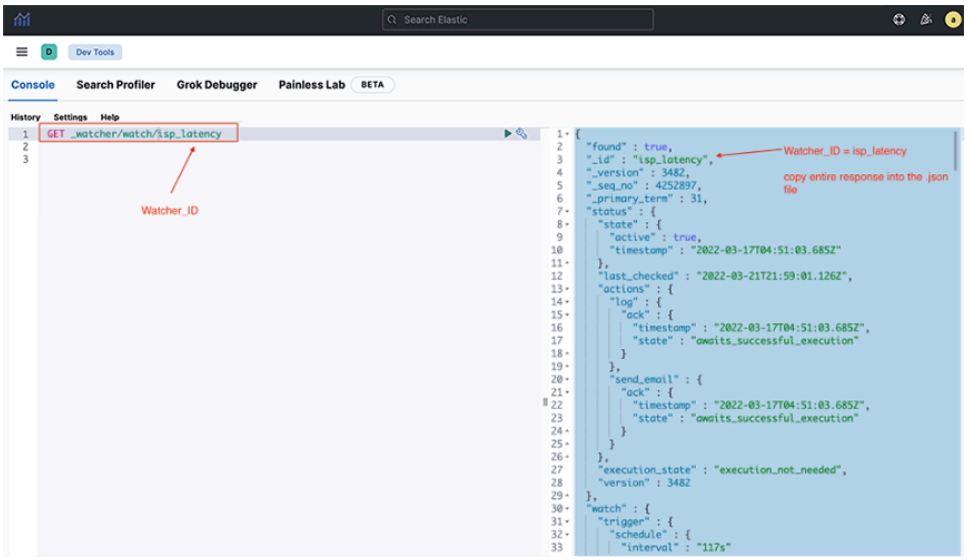
2. Open the main menu, then select **Dev Tools** > **Console**. Issue the **`GET`** API mentioned later with the **`watcher_id`**. The response appears in the output terminal.

   Run the following API call:

```
GET _watcher/watch/<watcher_id>
```

Replace ***Watcher_ID*** with the **`watcher_id`** name copied in **Step 1**.

Figure 10-6: GET API



3. Copy the API response from **Step 2** into a **`.json`** file with the terminology that suits the environment, and keep track of it. As an example, the following may be helpful to nomenclature: **`Arista_pod1_2022-02-03_isp_latency_watcher.json`**.

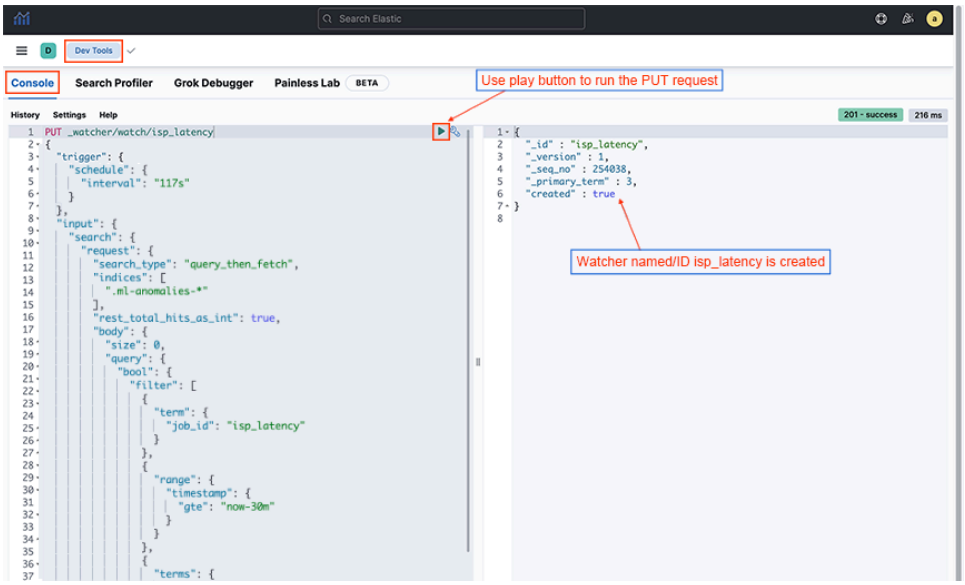
10.3.2 Importing Watchers

1. Not all exported fields are needed when importing a Watcher. To filter out the unwanted fields from the exported file, select the **jq** utility. Select **`jq .watch <exported_watcher.json>`**and import the output.

Figure 10-7: jq Command Output

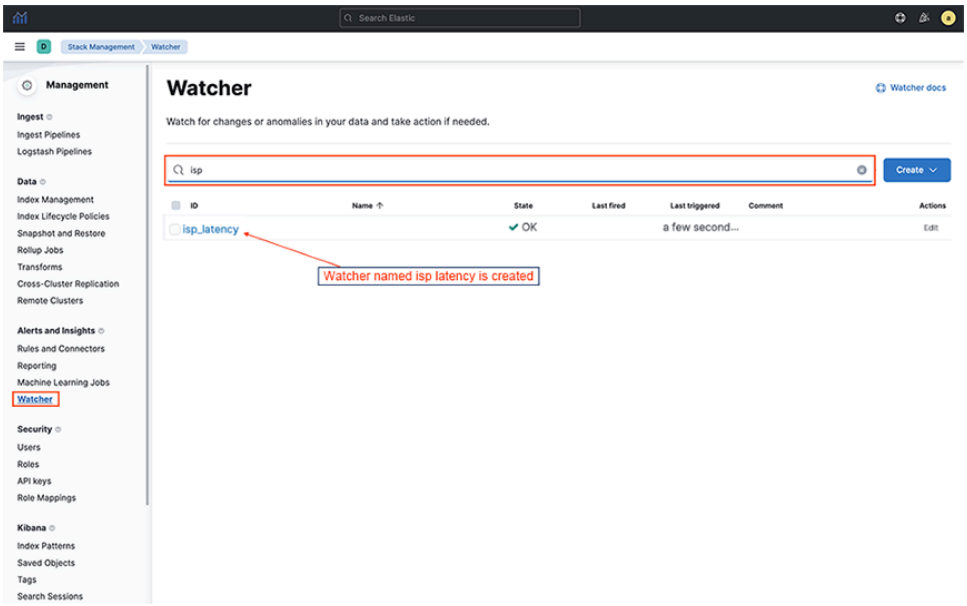
2. Click **DevTools** > **console**, enter the API **`PUT_watcher/watch/<watcher_id>`**, and copy the **Step 1** output into the following screen. Replace ***Watcher_ID*** with the desired Watcher name. The output terminal will confirm the creation of the Watcher.

Figure 10-8: PUT API in Dev Tools Console



3. Locate the newly created Watcher in the **Main menu** > **Management** > **Elasticsearch** > **Watcher** > **search with Watcher_ID**.

Figure 10-9: Watcher



10.4 Import and Export of Machine Learning Jobs

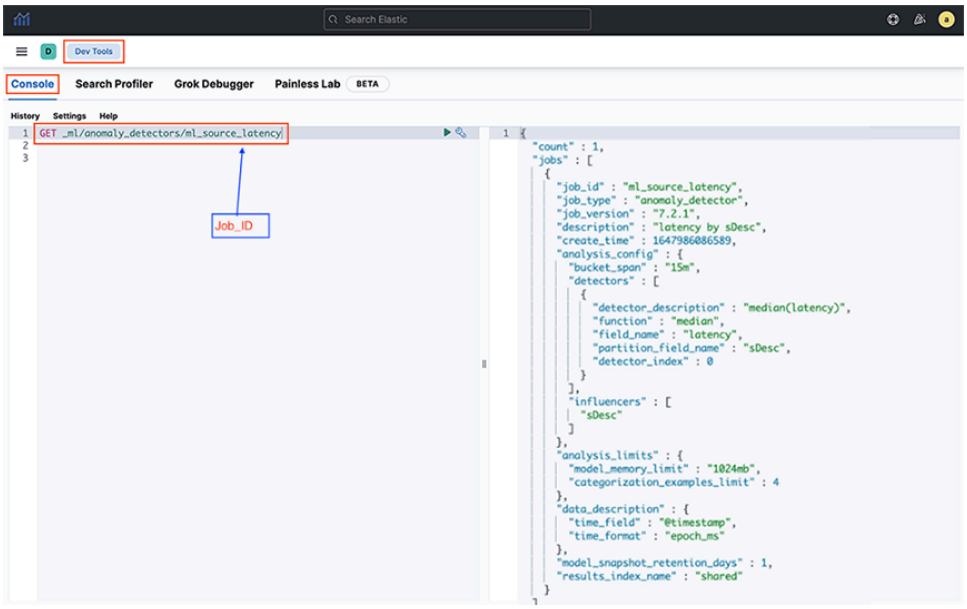
Machine Learning (ML) automates time series data analysis by creating accurate baselines of normal behavior and identifying anomalous patterns. This section explains ways to back up and restore the Machine Learning jobs in Arista Analytics.

10.4.1 Exporting Machine Learning Jobs

1. Open the main menu, then select **Dev Tools** > **Console**. Send a **`GET _ml/anomaly_detectors/<Job-id>`** request to Elasticsearch and view the response of all the Machine Learning anomaly jobs.

Replace ***Job_id*** with the ML job name. The system displays the following output when executing the **`GET`** request.

Figure 10-10: Main Menu > Dev Tools > Console



2. Copy the **`GET`** API response of the ML job into a **`.json`** file with terminology that suits your environment and keep track of it. An example of appropriate nomenclature might be **`Arista_pod1_2022-02-03_ML_Source_Latency_ML_job.json`**.

10.4.2 Importing Machine Learning Jobs

1. It is optional to import all the exported fields. Only ***description***, ***analysis_con#g***, and ***data_description*** fields may be needed. Running **`jq '.jobs[] |{description, analysis_config, data_description}'<json-filename>`** copies the output into the **Dev tools** console. Replace ***json-#lename*** with the filename of the JSON file previously exported.

   Run the following API call:

```
jq '.jobs[] |{description, analysis_config, data_description}' Arista_pod1_
2022-02-03_ML_Source_Latency_ML_job.json
```

Figure 10-11: jq Required Fields

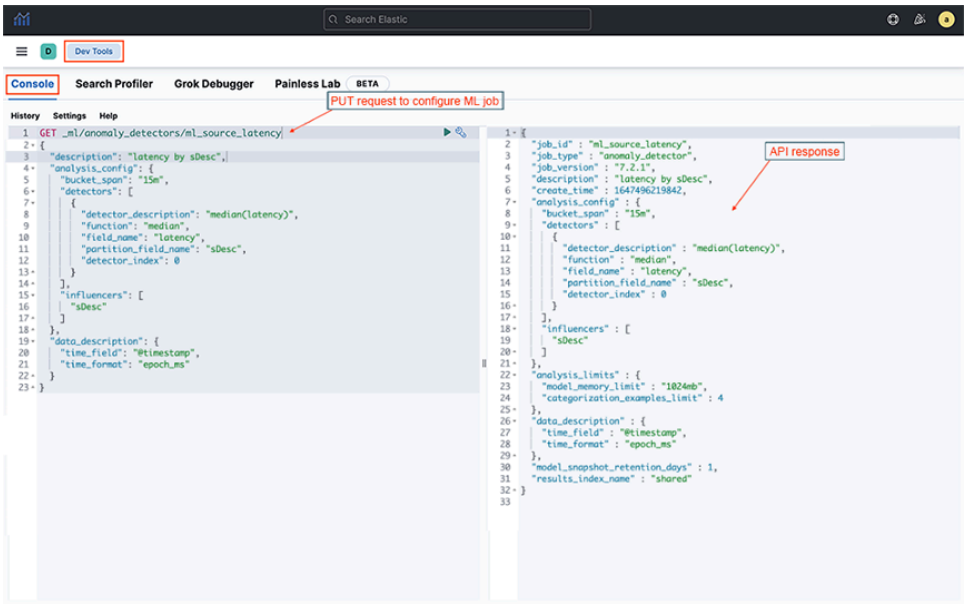
2. Select **Dev tools** > **Console** and copy the **Step 1** output into the following screen and the **`PUT`** request.

   Run the following API call:

   ```
   PUT _ml/anomaly_detectors/<ml_job_name>
   ```

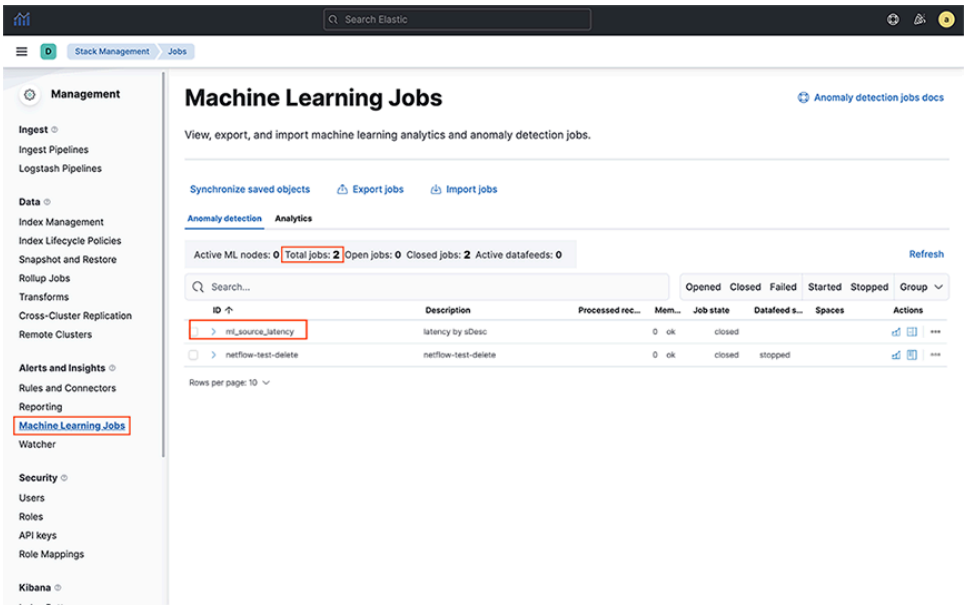
   Replace ***ml_job_name*** with the specific string of the ML Job name.

   Figure 10-12: PUT ML Jobs API



3. The successful response to the **`PUT`** request confirms the creation of the ML Job. Further, verify imported ML jobs by selecting **Main menu** > **Machine Learning** > **Job Management** > **search with ML Job Name**.

   Figure 10-13: ML Job Verification



Chapter 11

TACACS+ and RADIUS Control

This appendix describes using TACACS+ and RADIUS servers to control administrative access to the Analytics Node.

11.1 Using AAA Services with Arista Analytics

Select remote Authentication, Authorization, and Accounting (AAA) services using TACACS+ or RADIUS servers to control administrative access to the Analytics Node CLI.

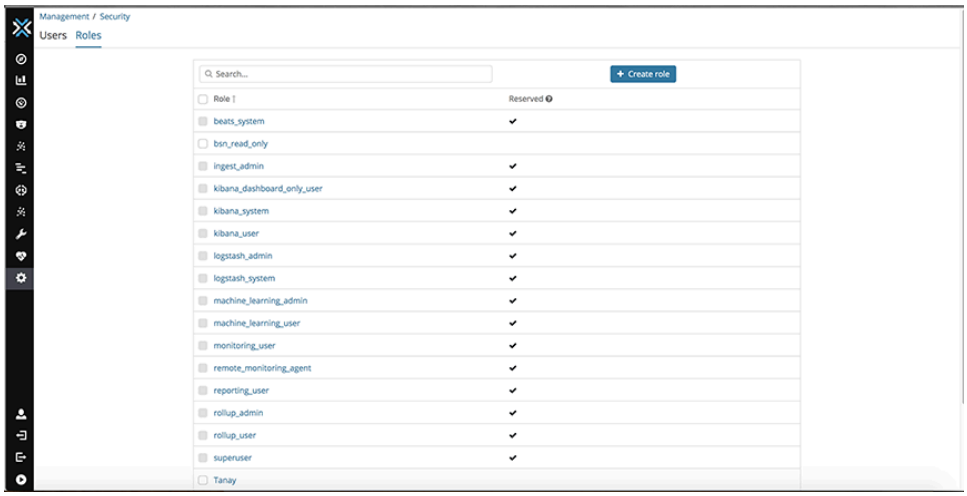
The following table lists the accepted Attribute-Value (AV) pairs:

Attributes	Values
BSN-User-Role	admin read-only bigtap-admin bigtap-read-only

Note: The remotely authenticated **admin** and **bigtap-admin** users and the **read-only** and **bigtap-read-only** users have the same privileges. The **bigtap-admin** and **bigtap-read-only** values are supported to create BMF-specific entries without affecting the **admin** and **read-only** TACACS+ server entries.

You must also create a role in Elasticsearch with the same name as the group configured in the CLI.

Figure 11-1: Creating a Group in Elasticsearch



A remotely authenticated admin user has full administrative privileges. Authenticate the **read-only** users on the switch remotely. Read-only access is not configurable for locally authenticated user accounts.

read-only users can only access login mode, from which they can view most **`show`** commands, with some limitations, including the following:

- TACACS, SNMP, and user configuration are not visible to the **read-only** user in the output from the show running-config command.
- **`show snmp`**, **`show user`**, and **`show support`** commands are disabled for the **read-only** user.

Note: Local authentication and authorization take precedence over remote authentication and authorization.

Privileges at the remote TACACS+ server must be configured using the following attribute-value pairs:

- **Supported attribute name**: BSN-User-Role
- **Supported attribute values**: admin, read-only

Select a TACACS+ server to maintain administrative access control instead of using the Analytics Node local database. However, it is a best practice to keep the local database as the secondary authentication and authorization method in case the remote server becomes unavailable.

11.1.1 DMF TACACS+ Configuration

The DANZ Monitoring Fabric (DMF) requires the following configuration on TACACS+ servers and the configuration required on the Analytics Node.

Authentication Method

- Configure the TACACS+ server to accept ASCII authentication packets. Do not select the **single connect-only** protocol feature.
- The DMF TACACS+ client uses the ASCII authentication method. It does not use PAP.

Device Administration

- Configure the TACACS+ server to connect to the device administration login service.
- Do not use a network access connection method, such as PPP.

Group Memberships

- Create a **bigtap-admin** group. Make all DANZ Monitoring Fabric users part of this group.
- TACACS+ group membership is specified using the **BSN-User-Role** AV Pair as part of TACACS+ session authorization.
- Configure the TACACS+ server for session authorization, not for command authorization.

Note: The **BSN-User-Role** attribute must be specified as **Optional** in the **`tac_plus.conf`** file to use the same user credentials to access ANET and non-ANET devices.

Enabling Remote Authentication and Authorization on the Analytics Node

Use the following commands to configure remote login authentication and authorization. The examples use the SSH default for connection type.

```
analytics-1# tacacs server host 10.2.3.201
analytics -1# aaa authentication login default group tacacs+ local
analytics -1# aaa authorization exec default group tacacs+ local
```

All users in the **bigtap-admin** group on TACACS+ server ***10.2.3.201*** have full access to the Arista Analytics Node.

User Lockout

Use the following command to lock out an AAA user after a calculated number of incorrect login attempts.

```
(config)#aaa authentication policy lockout failure F window W duration D
max-failures = F = [1..255] duration = D = [1..(2^32 - 1)] window = W = [1..(2^32 - 1)]
```

11.2 Adding a TACACS+ Server

To view the current TACACS+ configuration, enter the **`show running-config`** command, as in the following example:

```
analytics -1(config-switch)# show run switch BMF-DELIVERY-SWITCH-1 tacacs override-enabled
tacacs server host 1.1.1.1 key 7 020700560208
tacacs server key 7 020700560208
analytics -1(config-switch)#
```

It displays the TACACS+ key value as a type7 secret instead of plaintext.

Complete the following steps to configure the Analytics Node with TACACS+ to control administrative access to the switch.

Identify the IP address of the TACACS+ server and any key required for access using the **`tacacs server`** command, which has the following syntax:

```
tacacs server <server> [key {<plaintext-key> | 0 <plaintext-key> | 7 <encrypted-key>}
```

You can enable up to four AAA servers by repeating this command for each server. For example, using a plaintext key, the following command enables TACACS+ with the server running at ***10.2.3.4***.

```
analytics -1(config-switch)# tacacs server 10.1.1.1 key 0 secret
```

In case of a missing key, it uses an empty key.

Note: Do not use the pound character (#) in the TACACS secret. It is the start of a comment in the **`PAM config`** file.

Each TACACS+ server connection can be encrypted using a pre-shared key.

To specify a key for a specific host, use one of the following commands:

```
analytics -1# tacacs server host <ip-address> key <plaintextkey>
analytics -1# tacacs server host <ip-address> key 0 <plaintextkey>
analytics -1# tacacs server host <ip-address> key 7 <plaintextkey>
```

Replace ***plaintextkey*** with a password up to ***63*** characters in length. This key can be specified either globally or for each host. The first two forms accept a plaintext (literal) key, and the last form accepts a pseudo-encrypted key, such as that displayed with **`show running-config`**.

It uses the global key value when no key is specified for a given host. An empty key is assumed when no key is specified globally or specified for a given host.

The following example uses the **key 7** option followed by the encrypted string:

```
analytics-1(config-switch)# tacacs server 10.1.1.1 key 7 0832494d1b1c11
```

Note: Be careful while configuring TACACS+ to avoid disabling access to the Analytics Node.

11.3 Setting up a TACACS+ Server

Refer to your AAA server documentation for further details or instructions on setting up other servers.

After installing the TACACS+ server, complete the following steps to set up authentication and authorization for Analytics Node with the TACACS+ server:

1. Configure users and groups.
2. In the **`/etc/tacacs/tac_plus.conf`** file, specify the user credentials and group association.

```
# user details
user = user1 {
member = anet-vsa-admin
login = des a9qtD2JXeK0Sk
}
```

3. Configure the groups to use one of the AV pairs supported by the Analytics Node (for example, BSN-User-Role=admin for admin users).

```
# group details#
ANET admin group
group = anet-vsa-admin {
    service = exec {
    BSN-User-Role="admin"
    }
}
# BSN read-only group
group = anet-vsa-read-only {
    service = exec {
    BSN-User-Role="read-only"
    }
}
```

4. Configure the TACACS+ server and AAA on the Analytics Node.

```
tacacs server host <IP address> key server's secret>
aaa authentication login default group tacacs+ local
```

```
aaa authorization exec default group tacacs+ local
aaa accounting exec default start-stop locals group tacacs+
```

This configuration sets authentication and authorization to first connect to the TACACS+ server to verify user credentials and privileges. It checks the user account locally only when the remote server is unreachable. In this example, accounting stores audit logs locally and sends them to the remote server.

11.3.1 Credentials for the Analytics Node and Other Devices

To use the same user credentials for the Analytics Node and other devices, a specific setting in the **`tac_plus.conf`** file is necessary. Configure the ***BSN-User-Role*** attribute within the **`tac_plus.conf`** file as "Optional".

This configuration allows the system to correctly authenticate and authorize users across different types of network devices.The following is an example of the implementation setting in **`tac_plus.conf`**.

```
group = group-admin {
    default service = permit
    service = exec {
        optional BSN-User-Role = "admin"
    }
}
```

11.3.2 RBAC-based Configuration for Non-default Group User

To create a Role Based Access Control (RBAC) configuration for a user in a non-default group, complete the following steps:

1. Create a group ***AD1***.

   ```
   group AD1
   ```

   Do not associate with any local users.
2. Use the same group name on the TACACS+ server and associate a user to this group.

   Note: The attribute should be ***BSN-User-Role***, and the value should be the group name.

   The following is an example from the open TACACS+ server configuration.

   ```
   group = AD1 {
   service = exec {
   BSN-User-Role="AD1"
   }
   }
   ```
3. After you create the group, associate a user to the group.

   ```
   user = user3 {
   member = AD1
   login = cleartext user3
   ```
4. Click save.

11.4 Using RADIUS for Managing Access

RADIUS does not separate authentication and authorization. Be careful when authorizing a user account with a remote RADIUS server to use the password configured for the user on the remote server.

By default, authentication and authorization functions are set to local while the accounting function is disabled. The only supported privilege levels are as follows:

- **admin**: Administrator access, including all CLI modes and debug options.
- **read-only**: Login access, including most show commands.

Note: You cannot authenticate the **admin** and **recovery** user accounts remotely using TACACS. Always authenticate these accounts locally to prevent administrative access from being lost in case a remote AAA server is unavailable.

The **admin** group provides complete access to all network resources, while the **read-only** group provides read-only access to all network resources.

DANZ Monitoring Fabric also supports communication with a remote AAA server (TACACS+ or RADIUS). The following summarizes the options available for each function:

- **Accounting**: local, local and remote, or remote.
- **Authentication**: local, local then remote, remote then local, or remote.
- **Authorization**: local, local then remote, remote then local, or remote.

Note: Fallback to local authentication occurs only when the remote server is unavailable, not when authentication fails.

Privileges at the remote TACACS+ server must be configured using the attribute-value pairs shown in the following table:

Supported attribute names	Supported attribute values
BSN-User-Role	admin read-only bigtap-admin bigtap-read-only

The ***BSN-AV-Pair*** attribute sends CLI command activity accounting to the RADIUS server.

11.4.1 Adding a RADIUS Server

Use the following command to specify the remote RADIUS server:

```
radius server host <server-address> [timeout {<timeout>}][key {{<plaintext>} | 0 {<plaintext>} | 7
 {<secret>}}]
```

For example, the following command identifies the RADIUS server at the IP address ***192.168.17.101***:

```
analytics-1(config)# radius server host 192.168.17.101 key admin
```

You can enter this command up to five times to specify multiple RADIUS servers. The Analytics Node tries to connect to each server in the order they are configured.

11.4.2 Setting up a FreeRADIUS Server

After installing the FreeRADIUS server, complete the following steps to set up authentication and authorization for the Analytics Node with the RADIUS server:

1. Create the BSN dictionary and add it to the list of used dictionaries.

```
create dictionary /usr/share/freeradius/dictionary.bigswitch with the contents below:
VENDOR      Big-Switch-Networks 37538
BEGIN-VENDOR Big-Switch-Networks
ATTRIBUTE   BSN-User-Role   1         string
ATTRIBUTE   BSN-AVPair      2
            string
END-VENDOR Big-Switch-Networks
```

2. Include the **bigswitch** dictionary in the RADIUS dictionary file: **`/usr/share/freeradius/dictionary`**

```
$INCLUDE     dictionary.bigswitch
```

3. Configure a sample user with **admin** and **read-only** privileges.

   The following is an example that defines and configures a user, opens the user file **`/etc/freeradius/users`**, and inserts the following entries:

```
"user1"     Cleartext-Password := "passwd"
            BSN-User-Role := "read-only",
```

Note: It shows the VSA's association with the user and its privileges. A database and an encrypted password are necessary for an actual deployment.

   The following example authorizes the ***user2*** for RBAC group ***AD1***:

```
"user2"     Cleartext-Password := "passwd"
            BSN-User-Role := "AD1",
```

4. Configure the RADIUS server and AAA on the Analytics Node.

```
radius server host <IP address> key server's secret>
aaa authentication login default group radius local
aaa authorization exec default group radius local
aaa accounting exec default start-stop group radius local
```

   This configuration sets authentication and authorization to first connect to the RADIUS server to verify user credentials and privileges. AAA fallback to local occurs only when the remote server is unreachable. In this example, accounting stores audit logs locally and sends them to the remote server.

5. Add the Analytics Node subnet to the allowed subnets (**`clients.conf`**) on the RADIUS server.

   It is required when access to the RADIUS server is limited to allowed clients or subnets. The following is an example of the **`clients.conf`** file:

```
client anet {
    ipaddr = 10.0.0.0/8
```

```
    secret = <server's secret>
}
```

6. Restart the FreeRADIUS service on the server to enable the configuration.

   The following is an example accounting record sent from the Analytics Node to the RADIUS server after adding the ***BSN-AVPair*** attribute to the **`/usr/share/freeradius/dictionary.bigswitch`** file.

```
s
```

Appendix A

Watcher Alerts

The following appendix describes the procedure for creating Watcher alerts for machine learning jobs, emails, and remote Syslog servers.

A.1 Watcher Alert

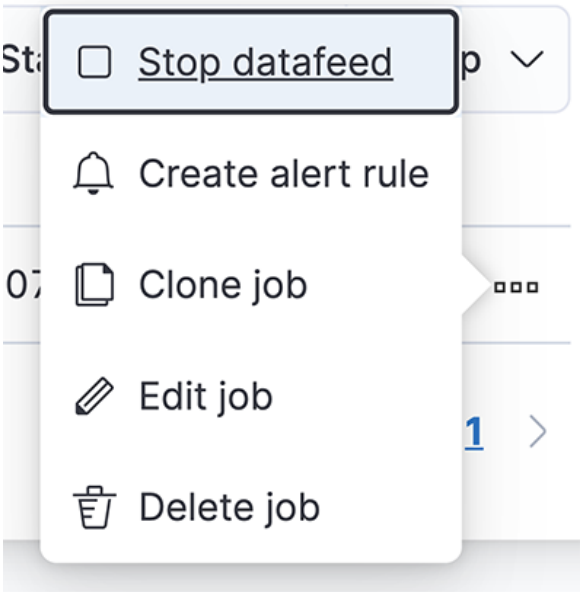
DMF 8.1 uses ***Elasticsearch 7.2.0***, where the inter-container functional calls are HTTP-based. However, ***DMF 8.3*** uses ***Elasticsearch version 7.13.0***, which now requires HTTPS-based calls. It would require an extensive change in the system calls used by the Analytics Node (AN), and engineering is working on this effort. Arista recommends the following workaround until the earlier fixes are released.

Workaround Summary:

- Create a Watcher manually using the provided template.
- Configure the Watcher to select the job ID for the ML job that needs to send alerts.
- Select ‘webhook’ as the alerting mechanism within the Watcher to send alerts to 3rd party tools like ‘Slack.’

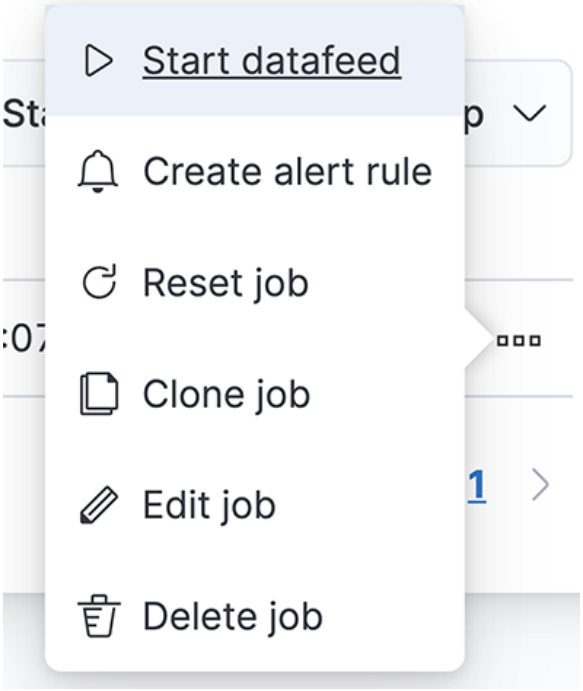
1. Access the AN's ML job page and click **Manage Jobs** to list the ML jobs.
2. If the data feed column shows as **stopped**, skip to **Step 3**. If it says **started**, click the **3 dots** for a particular ML job and **Stop** the data feed for the current ML job.

   Figure A-1: Stop Data Feed



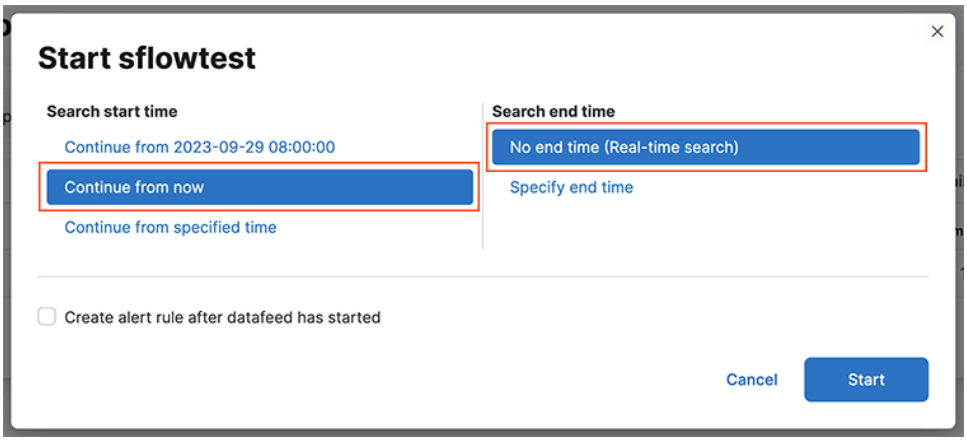
3. After the data feed has stopped, click the **3 dots** and start the data feed.

Figure A-2: Start Data Feed



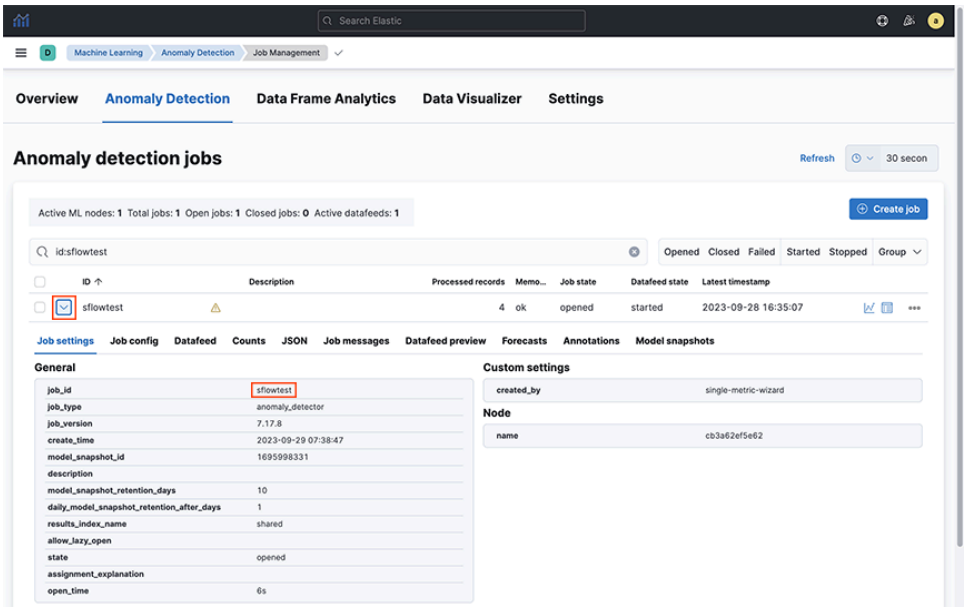
4. Select the options as shown in the following diagram.

Figure A-3: Job Time Options



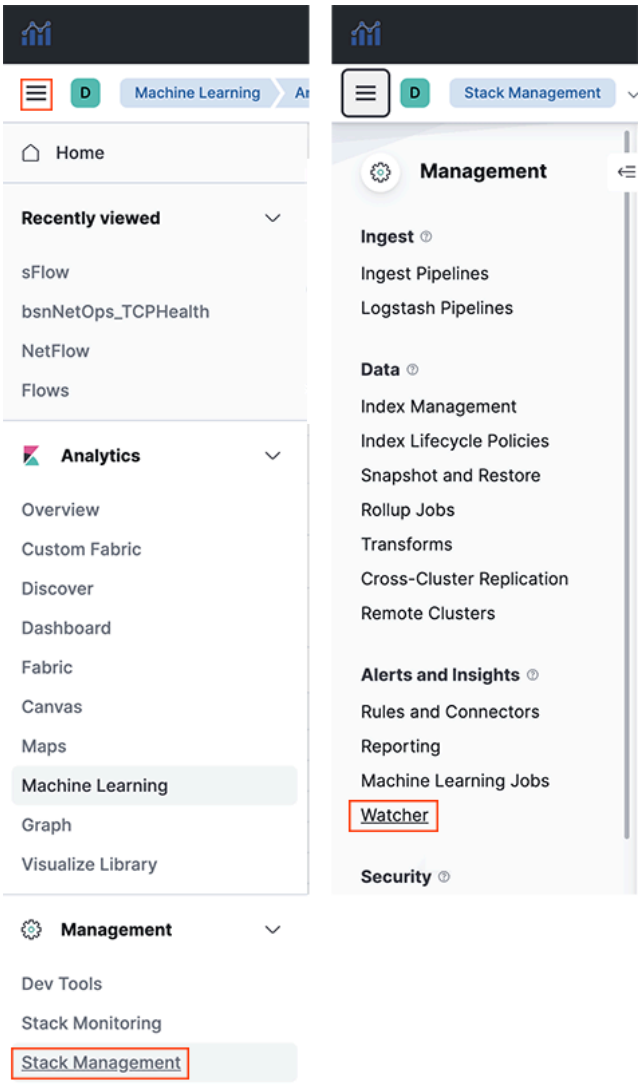
5. Confirm that the data feed has started. Note down the job ID of this ML job.

Figure A-4: ML Job Characteristics



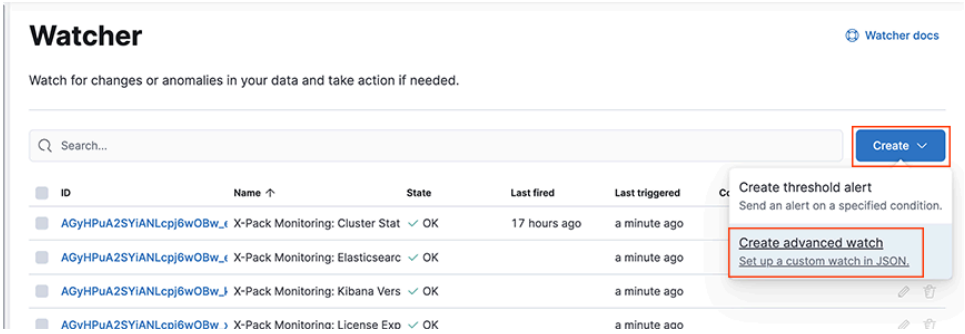
6. Access the **Watchers** page.

Figure A-5: Access Watchers



7. Create an advanced Watcher.

Figure A-6: Create Advanced Watcher



8. Configure the name of the Watcher (can include whitespace characters), e.g., ***Latency ML***.
9. Configure the ID of the Watcher (can be alphanumeric, but without whitespace characters), e.g., ***ml_latency***.

10. Delete the code from the **`Watch JSON`** section.
11. Copy and paste the following code into the Watcher. Replace the highlighted text according to your environment and your ML job parameters.

```
{
  "trigger": {
    "schedule": {
      "interval": "107s"
    }
  },
  "input": {
    "search": {
      "request": {
        "search_type": "query_then_fetch",
        "indices": [
          ".ml-anomalies-*"
        ],
        "rest_total_hits_as_int": true,
        "body": {
          "size": 0,
          "query": {
            "bool": {
              "filter": [
                {
                  "term": {
                    "job_id": "<use the id of the ML job retrieved in step 6.>"
                  }
                },
                {
                  "range": {
                    "timestamp": {
                      "gte": "now-30m"
                    }
                  }
                },
                {
                  "terms": {
                    "result_type": [
                      "bucket",
                      "record",
                      "influencer"
                    ]
                  }
                }
              ]
            }
          },
          "aggs": {
            "bucket_results": {
              "filter": {
                "range": {
                  "anomaly_score": {
                    "gte": 75
                  }
                }
              },
              "aggs": {
                "top_bucket_hits": {
                  "top_hits": {
                    "sort": [
                      {
                        "anomaly_score": {
                          "order": "desc"
                        }
                      }
                    ],
                    "_source": {
                      "includes": [
                        "job_id",
                        "result_type",
                        "timestamp",
                        "anomaly_score",
                        "is_interim"
                      ]
                    },
                    "size": 1,
                    "script_fields": {
```

```
                      "start": {
                        "script": {
                          "lang": "painless",
                          "source": "LocalDateTime.ofEpochSecond((doc[\"timestamp\"].val
ue.getMillis()-((doc[\"bucket_span\"].value * 1000)\n * params.padding)) / 1000, 0,ZoneOffset.
UTC).toString()+\":00.000Z\"",
                          "params": {
                            "padding": 10
                          }
                        }
                      },
                      "end": {
                        "script": {
                          "lang": "painless",
                          "source": "LocalDateTime.ofEpochSecond((doc[\"timestamp\"].val
ue.getMillis()+((doc[\"bucket_span\"].value * 1000)\n * params.padding)) / 1000, 0,ZoneOffset.
UTC).toString()+\":00.000Z\"",
                          "params": {
                            "padding": 10
                          }
                        }
                      },
                      "timestamp_epoch": {
                        "script": {
                          "lang": "painless",
                          "source": """doc["timestamp"].value.getMillis()/1000"""
                        }
                      },
                      "timestamp_iso8601": {
                        "script": {
                          "lang": "painless",
                          "source": """doc["timestamp"].value"""
                        }
                      },
                      "score": {
                        "script": {
                          "lang": "painless",
                          "source": """Math.round(doc["anomaly_score"].value)"""
                        }
                      }
                    }
                  }
                }
              }
            },
            "influencer_results": {
              "filter": {
                "range": {
                  "influencer_score": {
                    "gte": 3
                  }
                }
              },
              "aggs": {
                "top_influencer_hits": {
                  "top_hits": {
                    "sort": [
                      {
                        "influencer_score": {
                          "order": "desc"
                        }
                      }
                    ],
                    "_source": {
                      "includes": [
                        "result_type",
                        "timestamp",
                        "influencer_field_name",
                        "influencer_field_value",
                        "influencer_score",
                        "isInterim"
                      ]
                    },
                    "size": 3,
                    "script_fields": {
                      "score": {
                        "script": {
                          "lang": "painless",
                          "source": """Math.round(doc["influencer_score"].value)"""
                        }
```

```
                    }
                  }
                }
              }
            }
          },
          "record_results": {
            "filter": {
              "range": {
                "record_score": {
                  "gte": 75
                }
              }
            },
            "aggs": {
              "top_record_hits": {
                "top_hits": {
                  "sort": [
                    {
                      "record_score": {
                        "order": "desc"
                      }
                    }
                  ],
                  "_source": {
                    "includes": [
                      "result_type",
                      "timestamp",
                      "record_score",
                      "is_interim",
                      "function",
                      "field_name",
                      "by_field_value",
                      "over_field_value",
                      "partition_field_value"
                    ]
                  },
                  "size": 3,
                  "script_fields": {
                    "score": {
                      "script": {
                        "lang": "painless",
                        "source": """Math.round(doc["record_score"].value)"""
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.aggregations.bucket_results.doc_count": {
        "gt": 0
      }
    }
  },
  "actions": {
    "log": {
      "logging": {
        "level": "info",
        "text": "Alert for job [{{ctx.payload.aggregations.bucket_results.top_bucke
t_hits.hits.hits.0._source.job_id}}] at [{{ctx.payload.aggregations.bucket_results.top_bucke
t_hits.hits.hits.0.fields.timestamp_iso8601.0}}] score [{{ctx.payload.aggregations.bucket_resul
ts.top_bucket_hits.hits.hits.0.fields.score.0}}]"
      }
    },
    "my_webhook": {
      "webhook": {
        "scheme": "https",
        "host": "hooks.slack.com",
        "port": 443,
        "method": "post",
        "path": "<path for slack>",
        "params": {},
```

```
        "headers": {
          "Content-Type": "application/json"
        },
        "body": """{"channel": "#<slack channel name>", "username": "webhookbot", "text":"Alert
 for job [{{ctx.payload.aggregations.bucket_results.top_bucket_hits.hits.hits.0._source
.job_id}}] at [{{ctx.payload.aggregations.bucket_results.top_bucket_hits.hits.hits.0.fields.
timestamp_iso8601.0}}] score [{{ctx.payload.aggregations.bucket_results.top_bucket_hits.hits.h
its.0.fields.score.0}}]", "icon_emoji": ":exclamation:"}"""
      }
    }
  }
}
```

12. Click **Create Watch** to create the Watcher.

A.2 Kibana Watcher for Webhook Connector

This document specifically describes how to configure Watcher for Webhook-type connectors.

Kibana connectors in the **Stack Management** > **Alerts and Insights** provide seamless integration between the Elasticsearch alerting engine and external systems.

They enable automated notifications and actions to be triggered based on defined conditions, enhancing monitoring and incident response capabilities. Webhook connectors allow alerts to be forwarded to platforms like Slack and Google Chat, delivering customizable payloads to notify relevant teams when critical events occur.

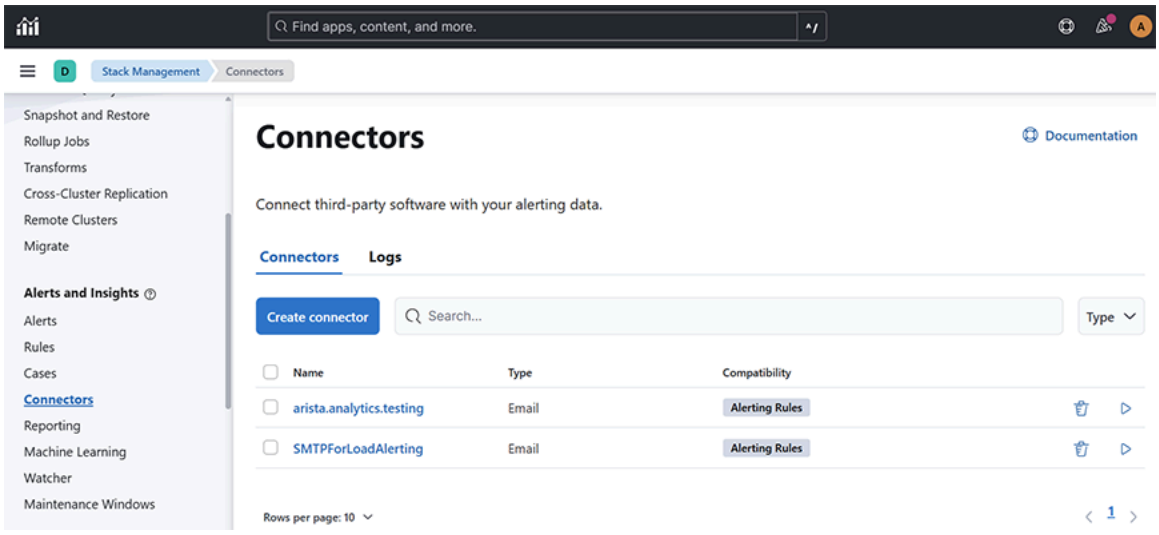
Configuring a Kibana Email Connector

The DMF 8.7.0 release supports the following connector integrations:

- Gmail via the Email Connector
- Google Chat and Slack via the Webhook Connector.

Select an existing Kibana email connector to send email alerts or create a connector by navigating to **Stack Management** > **Alerts and Insights** > **Connectors** > **Create Connectors**. Complete the following steps:

Figure A-7: Rules and Connectors

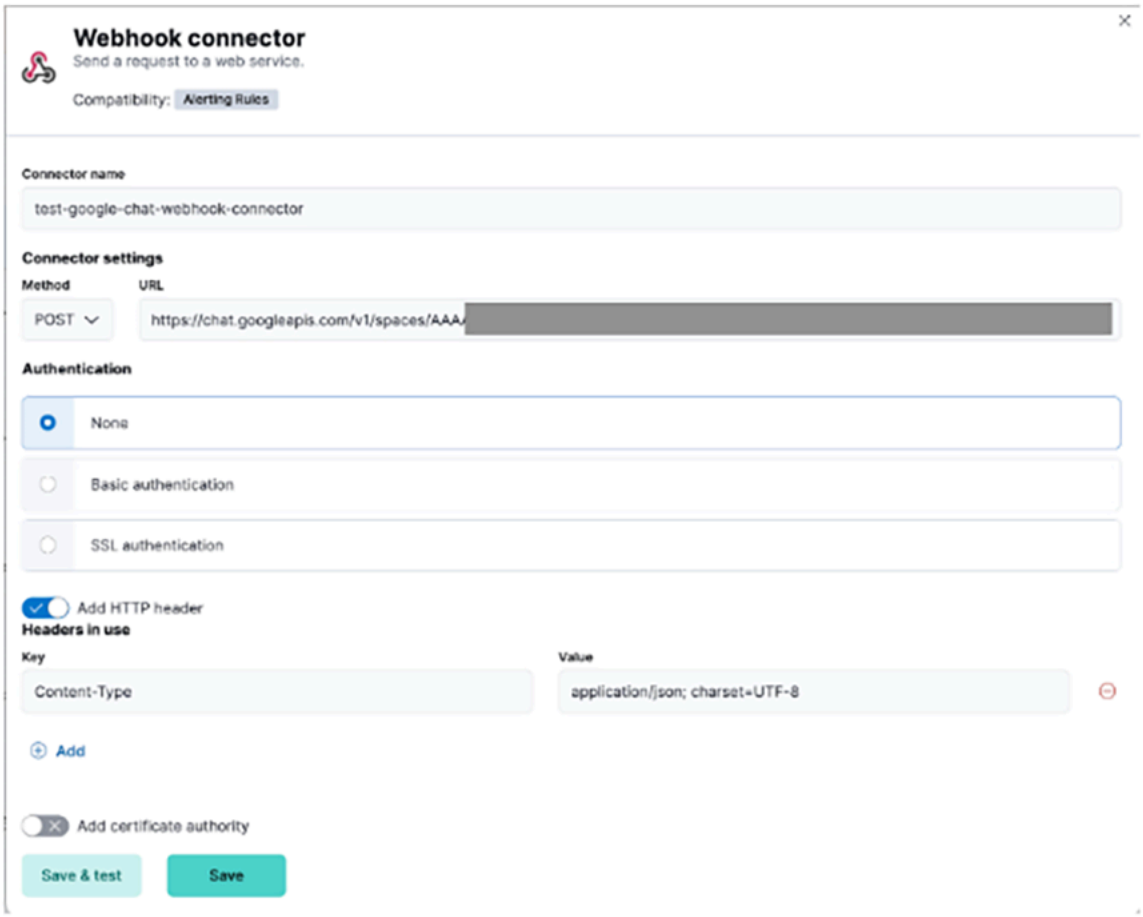


Note: Only the **Email** and **Webhook** connector types are available for DMF 8.7 release.

Google Chat Webhook Connector

HTTP header **Content-Type** is mandatory with the value ***application/json; charset=UTF-8***.

Figure A-8: Webhook Connector



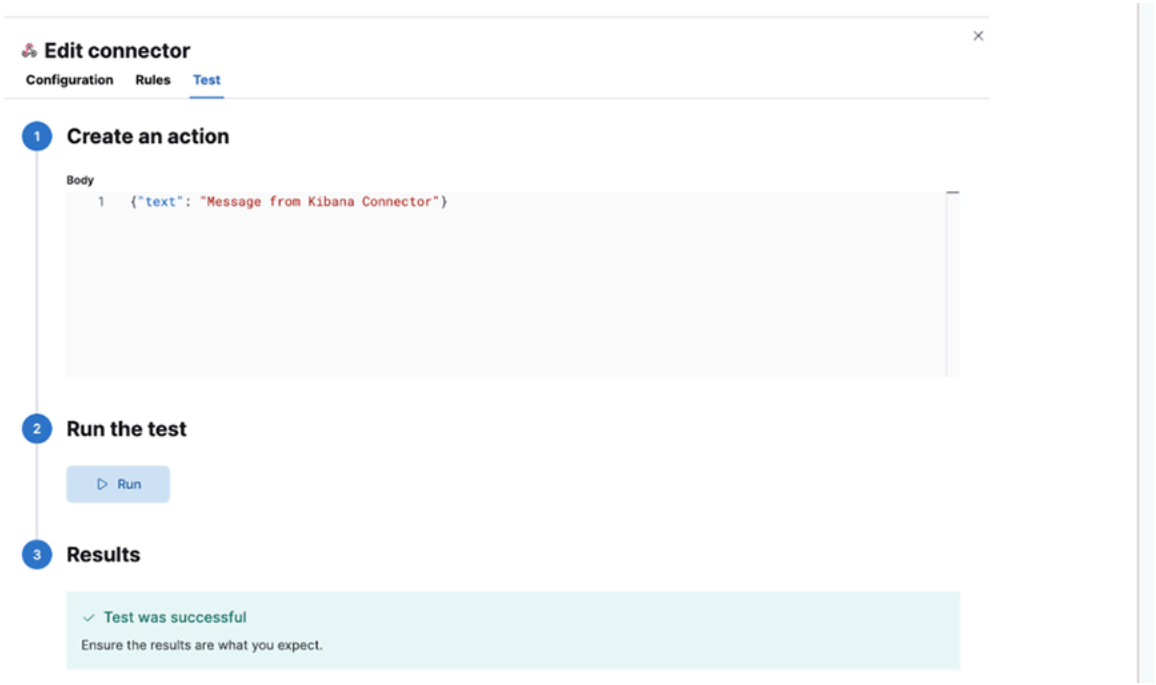
Under **Test** > **Create an action** > **Body** enter the following test:

```
{"text": "Message from Kibana Connector"}
```

Note: ***text*** field is mandatory in the **Body** section.

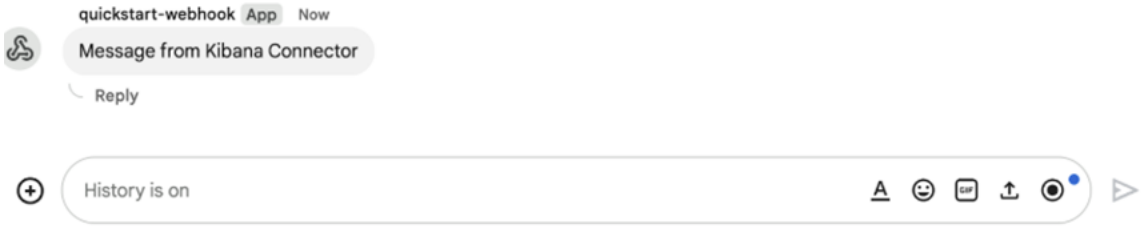
Ensure the result is successful and confirm that the respective chat window receives the message.

Figure A-9: Editing Google Connector to create action



The screen shows the following message.

Figure A-10: Google Chat Message



For any additional details, refer to https://developers.google.com/workspace/chat/quickstart/webhooks#create-webhook.

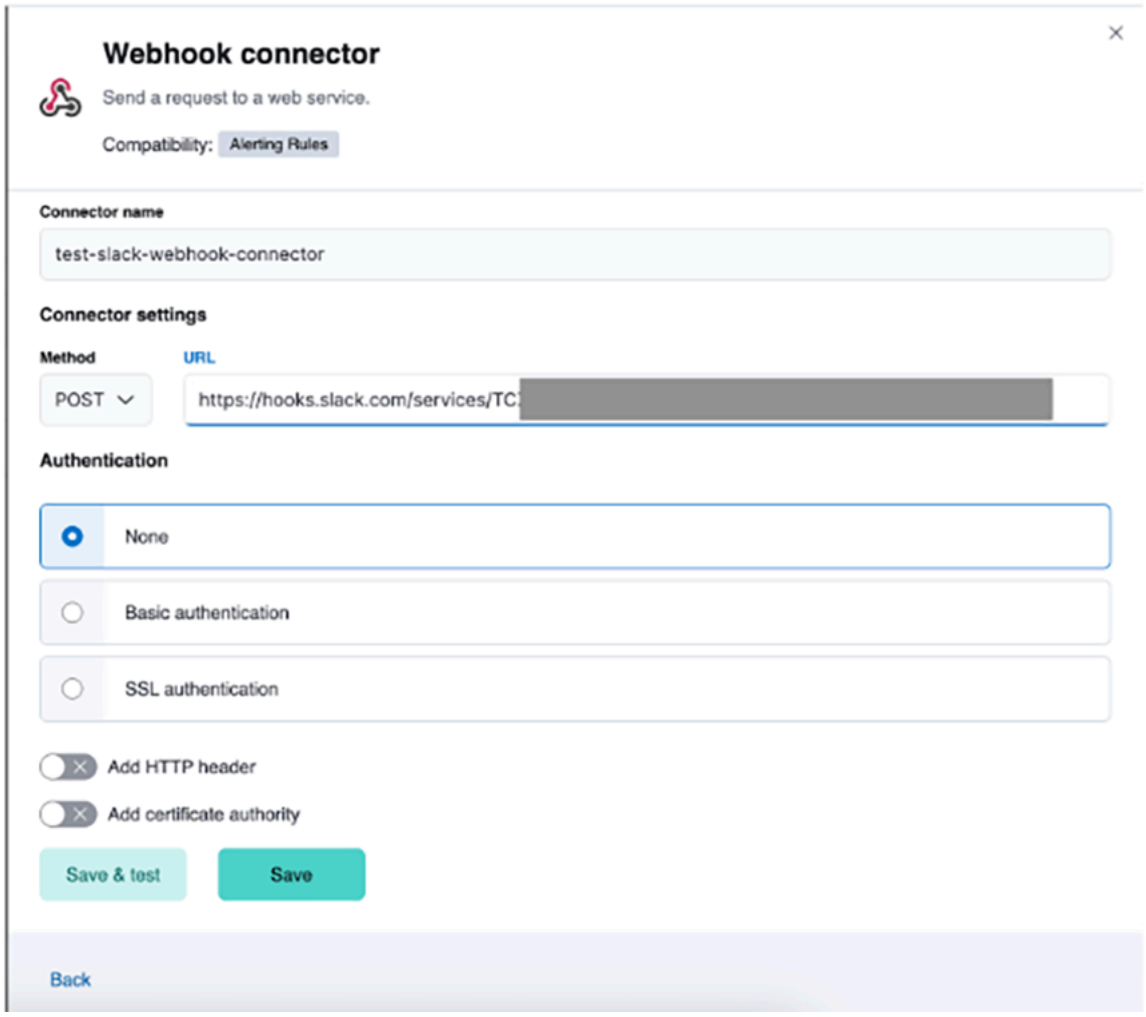
Slack Webhook Connector

Under **Test** > **Create an action** > **Body** enter the following test:

```
{"text": "Message from Kibana Connector"}
```

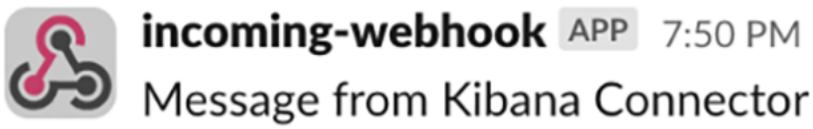
Note: The ***text*** field is mandatory in the **Body** section.

Figure A-11: Editing Slack Connector to create action



The screen shows the following message.

Figure A-12: Slack Chat Message



For any additional details, refer to https://api.slack.com/messaging/webhooks#getting_started.

Configuring a Watch

Configure a Watch using the **Watcher** or **Create** > **Create advanced watch** option.

A sample watcher script is as follows with an action type of Google Chat and Slack using the Webhook connectors.

```
"webhook_googlechat": {
     "webhook": {
       "scheme": "http",
       "host": "169.254.16.1",
       "port": 8000,
       "method": "post",
       "params": {},
       "headers": {},
       "body": """{"request_body": "{\"text\": \"The Elasticsearch cluster status is {{ctx.payload
.status}}.\"}","kibana_webhook_connector": "<google-chat-webhook-connector-name>"}"""
     }
```

- Add a webhook action with the following fields to select Google Chat and Slack Webhook Connectors.
  - **Method**: POST
  - **Scheme**: HTTP
  - **Host**: 169.254.16.1
  - **Port**: 8000
  - Specify the **Body** field as follows:
    - **kibana_webhook_connector**: The name of the Kibana connector (of type webhook) in string format. It is case-sensitive.
    - **request_body**: Enter the required fields: The connector gets the HTTP ***request body*** (in string format as text).

      Note: request_body value is specific to the connector's specification.

Click the **Save** button.

Google Chat Watcher configuration

Google Chat webhook accepts JSON formatted body with a mandatory ***text*** field. Hence, ***request body*** must have a JSON formatted string with text field. For example,

```
"{\"text\": \"The Elasticsearch cluster status is {{ctx.payload.status}}.\"}"
```

A sample watcher script with action type Google Chat Webhook is:

```
{
 "trigger": {
   "schedule": {
     "interval": "1h"
   }
 },
 "input": {
   "http": {
     "request": {
       "scheme": "https",
       "host": "10.10.10.10",
       "port": 443,
       "method": "get",
       "path": "/es/api/_cluster/health",
       "params": {},
       "headers": {
         "Content-Type": "application/json"
       },
       "auth": {
         "basic": {
           "username": "admin",
```

```
          "password": "::es_redacted::"
        }
      }
    }
  }
 },
 "condition": {
   "script": {
     "source": "ctx.payload.status != 'green'",
     "lang": "painless"
   }
 },
 "actions": {
   "webhook_googlechat": {
     "webhook": {
       "scheme": "http",
       "host": "169.254.16.1",
       "port": 8000,
       "method": "post",
       "params": {},
       "headers": {},
       "body": """{"request_body": "{\"text\": \"The Elasticsearch cluster status is {{ctx.payload
.status}}.\"}",  "kibana_webhook_connector": "BSN-Analytics-AppTest-GH-connector"}"""
     }
   }
 }
}
```

Slack Watcher configuration

Slack webhook accepts JSON formatted body with fields:

- **text**: (mandatory)
- **channel** : (optional) The channel name must match the same Slack channel for which the webhook is enabled.
- **username**: (optional) You can select any username.

For Example:

```
"{\"channel\": \"test-channel\", \"username\": \"webhookbot\", \"text\": \"The Elasticsearch
 cluster status is {{ctx.payload.status}}.\"}"
```

```
{
 "trigger": {
   "schedule": {
     "interval": "1m"
   }
 },
 "input": {
   "http": {
     "request": {
       "scheme": "https",
       "host": "10.10.10.1",
       "port": 443,
       "method": "get",
       "path": "/es/api/_cluster/health",
       "params": {},
       "headers": {
         "Content-Type": "application/json"
       },
       "auth": {
         "basic": {
           "username": "admin",
           "password": "::es_redacted::"
         }
       }
     }
   }
 },
 "condition": {
   "script": {
```

```
      "source": "ctx.payload.status != 'green'",
      "lang": "painless"
    }
  },
  "actions": {
    "webhook_slack": {
      "webhook": {
        "scheme": "http",
        "host": "169.254.16.1",
        "port": 8000,
        "method": "post",
        "params": {},
        "headers": {},
        "body": """{"request_body": "{\"channel\": \"test-webhook\", \"username\": \"webhookbot\
", \"text\": \"The Elasticsearch cluster status is {{ctx.payload.status}}.\"}",  "kibana_webho
ok_connector": "test-slack-webhook-connector"}"""
      }
    }
  }
}
```

Troubleshooting

If the Watcher is not sending an alert to the configured Webhook connector when the condition is already there:

- Check Kibana Connector, which is a type of **Webhook**.
- Check that the Kibana Connector is properly configured by running tests from UI. Check connector-specific configurations.
- Check for properly configured Watcher's trigger conditions.
- Make sure all required parameters are present in the connector's watcher configuration.
- To debug execution in CLI:
  - SSH to AN node
  - Log in to CLI mode command: **`debug bash`**
  - Review logs in **`/var/log/analytics/webhook_listener.log`** for any clues. Command: **`tail -f /var/log/analytics/webhook_listener.log`**
  - To execute service in debug mode:
    - Login as root command: **`sudo su`**
    - Stop service command: **`service webhook-listener stop`**.
    - Edit web-service in your preferred editor and set the logger to debug mode command: **`vi /usr/lib/python3.9/site-packages/webhook_listener/run_webhook_listener.py`** Change Line: **`LOGGER.setLevel(logging.INFO)`** to **`LOGGER.setLevel(logging.DEBUG)`**.
    - Start service command: **`service webhook-listener start`**.

You will see debug messages in the log file.

Note: After debugging, revert the debug level to info. If the problem persists, then raise a support ticket.

Limitations

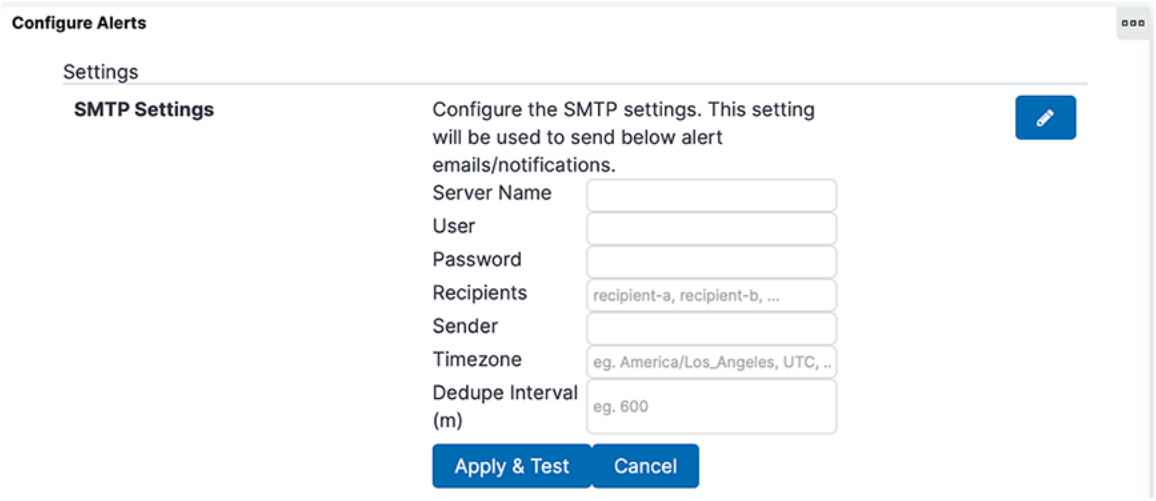
None.

A.3 Enabling Secure Email Alerts through SMTP Setting

Refresh the page to view the updated **SMTP Settings** fields.

The following is an example of the UI SMTP Settings in previous releases:

Figure A-13: SMTP Setting

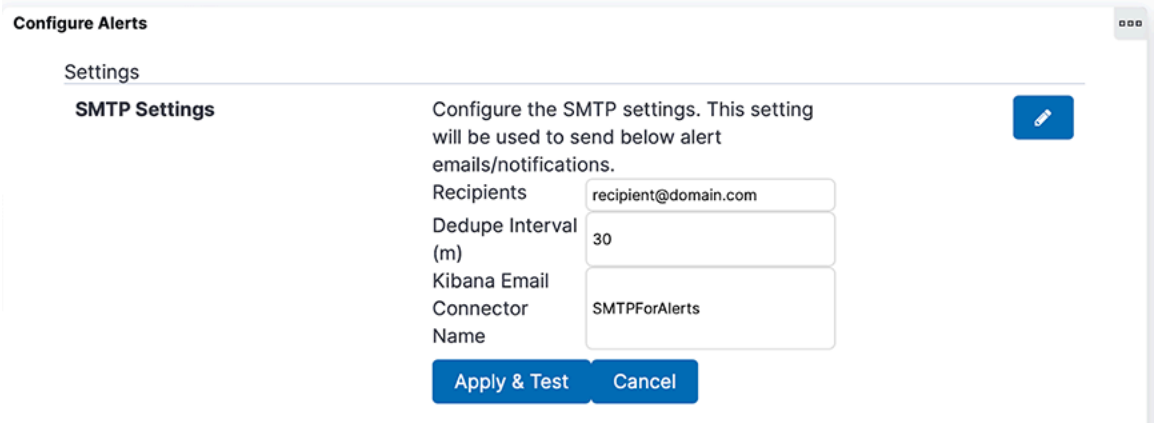


After upgrading the Analytics Node from an earlier version to the ***DMF 8.6***.* version, the following changes apply:

- Server **Name**, **User**, **Password**, **Sender**, and **Timezone** no longer appear in the SMTP Settings.
- A new field, **Kibana Email Connector Name**, has been added to SMTP Settings.
- The system retains **Recipients** and **Dedupe Interval** and their respective values in SMTP Settings.
- If previously configured SMTP settings exist:
  - The system automatically creates a Kibana email connector named **SMTPForAlerts** using the values previously specified in the fields Server Name, User (optional), Password (optional), and Sender.
  - The **Kibana Email Connector Name** field automatically becomes **SMTPForAlerts**.

The following settings appear in the UI after the upgrade to the ***DMF 8.6***.* version:

Figure A-14: Upgraded SMTP Setting



Troubleshooting

When **Apply & Test**, do not send an email to the designated recipients; verify the recipient email addresses are comma-separated and spelled correctly. If it still doesn’t work, verify the designated Kibana email connector matches the name of an existing Kibana email connector. Test that connector by navigating to **Stack Management** > **Rules and Connectors** > **Connectors**, selecting the connector's name, and sending a test email in the **Test** tab.

Appendix B

References

B.1 Related Documents

The following documentation is available for ***Arista Analytics 8.8.0***:

- ***Arista Analytics User Guide***
- ***Arista Analytics Deployment Guide***