Arista Analytics User Guide

Arista Networks

www.arista.com

Arista Analytics User Guide

DOC-06775-01

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

Arista Analytics Basic Operations

This chapter describes how to use Arista Analytics to monitor and analyze traffic and events in the monitoring fabric and the DANZ Monitoring Fabric controller. This chapter includes the following sections:

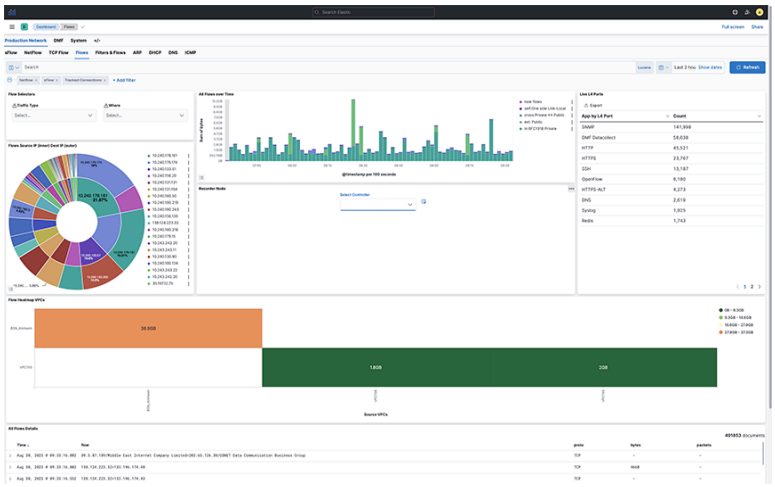
1.1 Overview

Arista Analytics provides a non-proprietary extensible UI that integrates DMF Recorder Nodes, DMF Service Nodes, and the DANZ Monitoring Fabric controlled using an SDN Controller. The system has an extensive library of visualization components and analytics to compose new dashboards and answer further questions as they arise. The Arista Analytics node/cluster answers questions that would otherwise require specialized applications, such as Application Data Management (ADM) or Intrusion Protection Management (IPM).
The Analytics node/cluster creates a document for each packet received and adds metadata regarding the context, including the time and the receiving interface, which ElasticSearch can use to search the resulting documents quickly and efficiently.

1.2 Flows Dashboard

The Flows dashboard is displayed when accessing Arista Analytics, as shown in the following figure.

Figure 1-1: Production Network > Flows Dashboard



The left panel provides the following options to access Arista Analytics features:

- **Fabric**: The home page for Analytics provides a series of tabs and sub tabs.
- **Controller**: Opens the DANZ Monitoring Fabric GUI on the controller identified using the **System** > **DMF controller** option.
- **Discover**: Use predefined indices to filter and display specific events.
- **Visualize**: Customize the graphics displays provided by Arista Analytics.
- **Dashboard**: Displays dashboards for DANZ Monitoring Fabric events.
- **Timelion**: Display events and other results according to time series.

Based on ElasticSearch, the Analytics GUI and most of its features and operations are documented in the Kibana documentation, available at the following URL:

https://www.elastic.co/guide/en/kibana/7.2/index.html

Kibana 7.2 is the version used for ***Arista Analytics version 7.3***.

1.3 Arista Analytics Fabric View

The Arista Analytics Fabric View displays in the following three tabs:

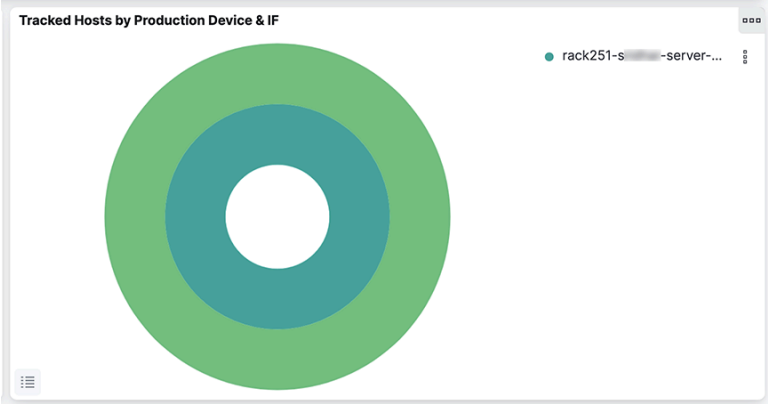
- **Production Network**: View information about the production network.
- **DMF Network**: View information about the monitoring network.
- **System**: Manage system configuration settings.

Each page contains panels with different functions and features. The network panels provide visualizations, such as pie charts, line graphs, or other graphic displays that reflect the current dashboard contents based on the specified query. The bottom panel lists all the events that match the current query. A pop-up window provides additional details about the selection when mousing over a panel.

1.3.1 Using Two-ring (by Production Switch) Pie Charts

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



When a second ring appears in a pie chart, click on any segment in the inner ring, and the outer ring provides a summary of information about the selected segment.

For example, in the **Tracked Hosts by Production Device & IF** pie chart above, the outer ring shows hosts tracked on each interface, while the inner ring summarizes the tracked hosts on each switch. Clicking on a segment for a specific switch on the inner ring filters the outer ring to show the tracked hosts for the interfaces on the selected switch.

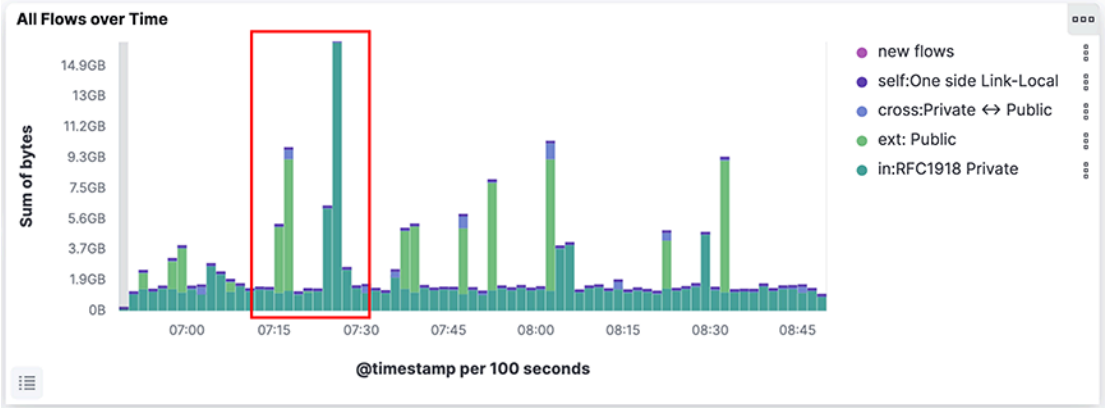
1.3.2 Filtering Information on a Dashboard

Filter the events displayed on a dashboard by dragging the mouse cursor around an area on the dashboard. This action limits the information displayed on the dashboard to events similar to those selected. Click on any pie chart slice to limit the display to the specific activity chosen. To change the color assigned to a specific protocol or other object, click the label on the list to the right of the chart.

1.3.3 Selecting the Time Range

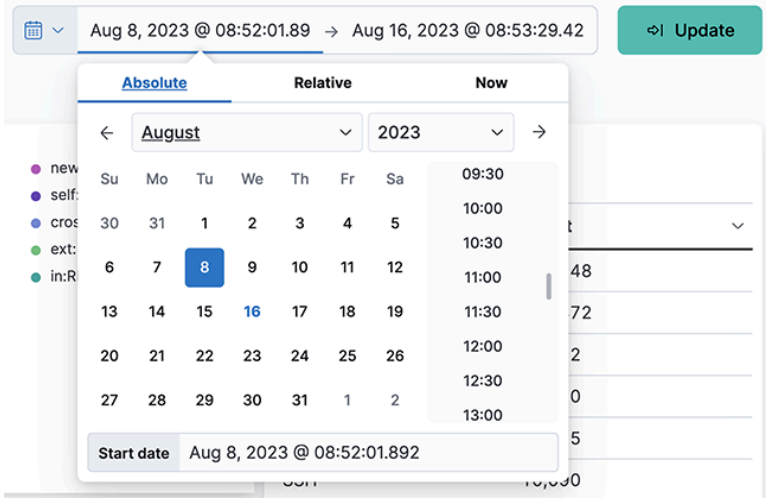
To restrict the current content to events occurring in a specific time period, click the mouse and drag it to surround the area on a time visualization, such as the Flows Over Time.

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

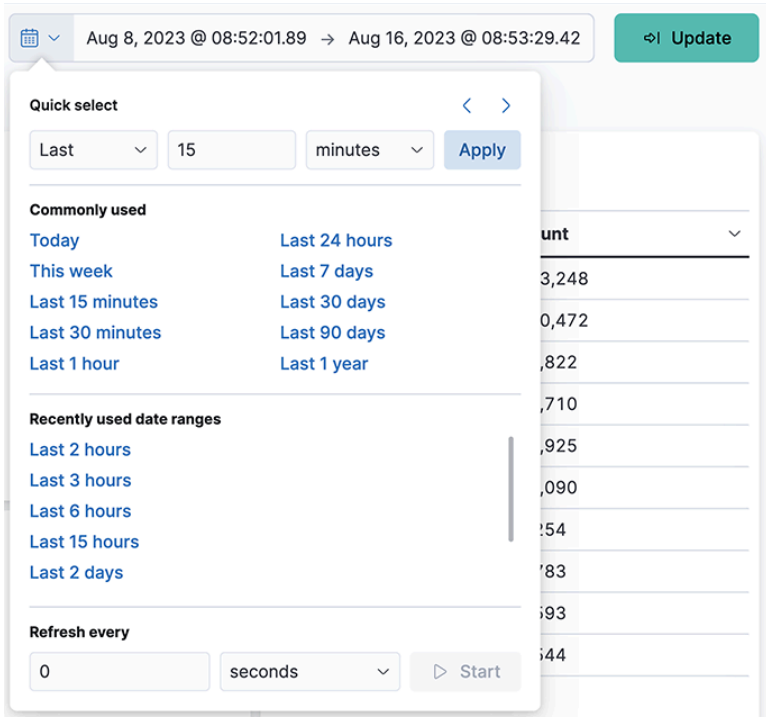


This dialog provides the following options for setting the time range:

- **Quick**: Simple settings, such as Today, Last 1 hour, and so forth.
- **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

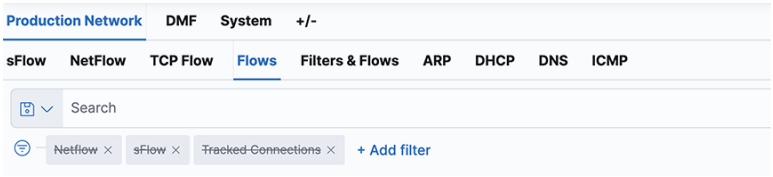


Select the refresh interval from the options provided. Click **Start** to disable the auto-refresh function.

1.3.4 Using the Search Field

The search field at the top of the dashboard lets you filter the current displays by any text or numbers that you type into the field.

Figure 1-6: Search Field



The green bars under the **Search** field show the currently applied filters. When you mouse over a green bar, a series of icons are displayed that let you 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.

When you click a segment on a pie chart, the appropriate filter is automatically inserted 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, if you enter the first part of an IP address, the displays will be updated to show only those IP addresses that match the characters entered. The following are some of the most useful search filters:

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

You can 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.

1.3.5 Search Performance Limitations

Do not execute a general query for 7 or 30 days. A 7 or 30 day query should be used with specific criteria, like querying a specific flow, filter interface, or DNS server.

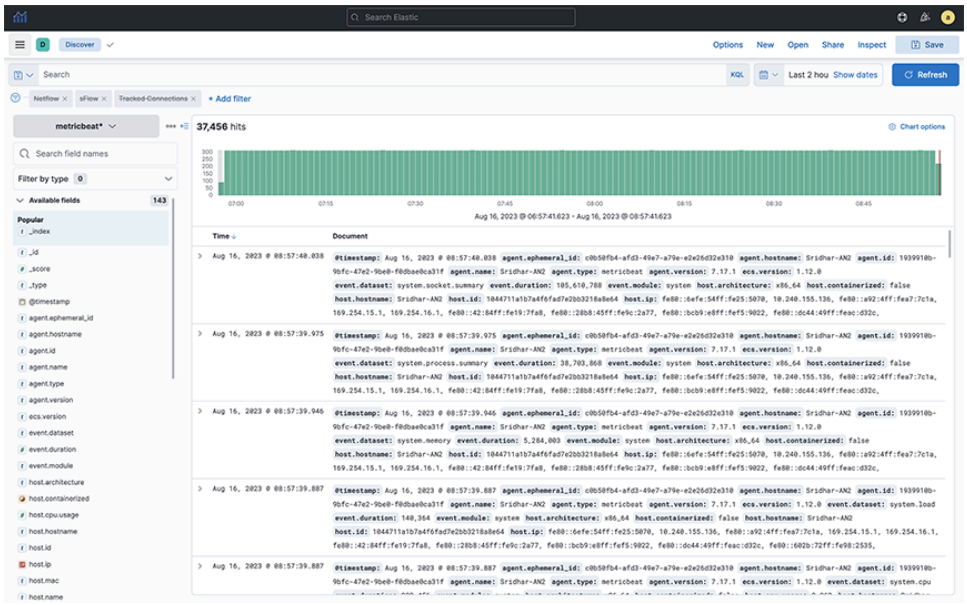
To query NetFlow or sFlow for longer periods, use the **FLOW** dashboard to determine the trend and then do a specific query, such as querying a specific flow or time period, on the Netflow or sFlow dashboard.

If you have a great deal of NetFlow traffic, use one Analytics node only for NetFlow and another node for other Analytics traffic.

1.4 Using Discover Mode

When you select the **Discover** option in the left panel of the **Analytics** window, the system displays the following page.

Figure 1-7: Discover Mode

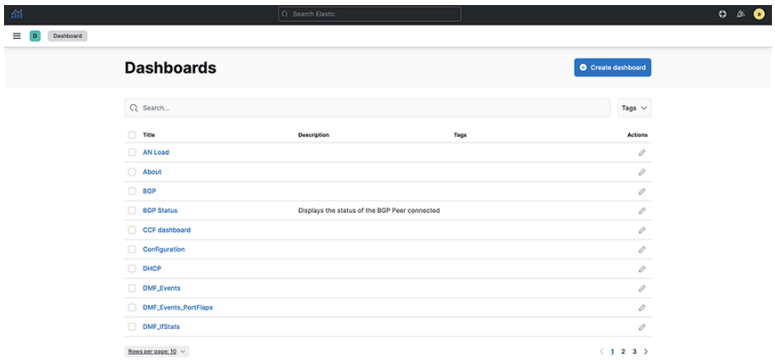


You can use Discover mode to see the indices available in the ElasticSearch database, and to identify the kind of data that is available.

1.5 Managing Dashboards

To manage dashboards, select the **Dashboards** option from the left panel on the **Analytics** window. 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/7.13/index.html

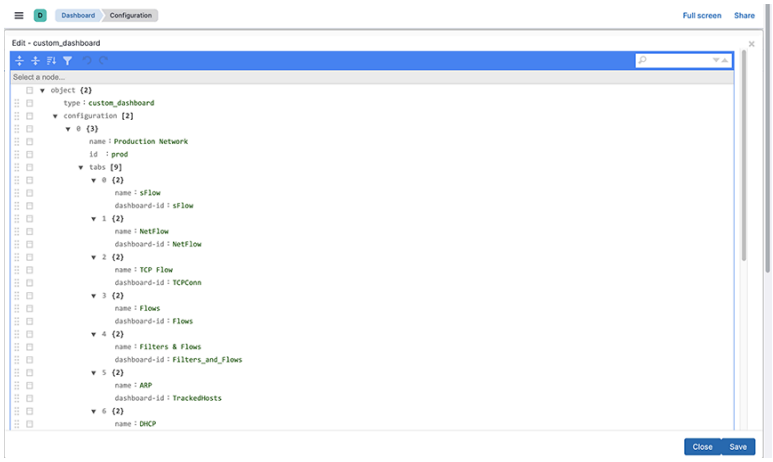
Note: Recommended Best Practices - While creating a dashboard or any saved objects, use the naming convention that suits your environment. For example, use a prefix to identify the content of the dashboard, and then use the body of the dashboard name to identify the type of dashboard. For instance, in the above screenshots, we used a naming pattern, prefixed with “ARISTA” and

specifying Type: dashboard allows a manageable set of things to individually appear to click or select all. Furthermore, exporting dashboards individually based on their type is a more appropriate option as modifications to a dashboard can be better tracked by this method. To reduce the potential for conflict in upgrades, your dashboards should use only visualizations and searches that you create; do not depend on default objects that might change in the upgrade.

1.6 Custom Dashboards

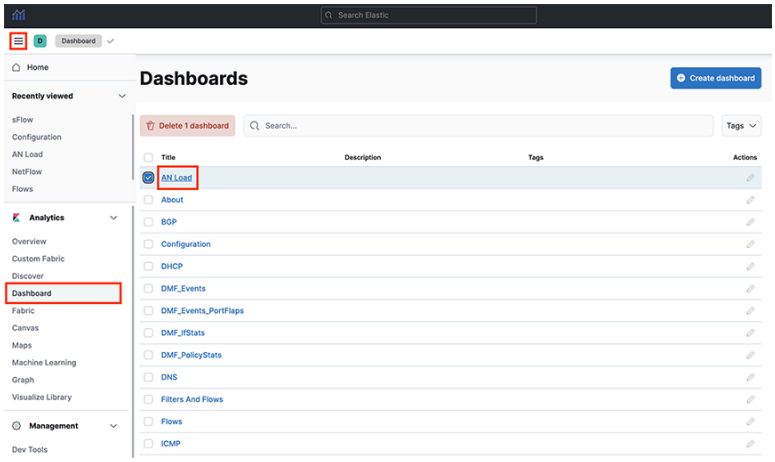
User can add or insert their own custom dashboard. select the **Dashboards** option from the left panel on the **Analytics** window. The system displays the following page, which is the default dashboard:

Figure 1-9: Default Dashboard Mode



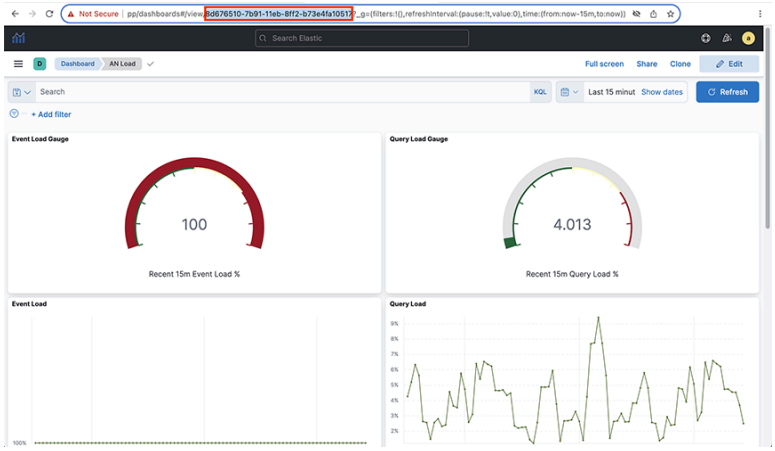
In the default dashboard, user can select the option, which one to customize as per their requirement.

Figure 1-10: Search for Dashboard



For the customization of the option on the dashboard, copy its **ID** as following.

Figure 1-11: Select the option and copy the ID



Note: ID has to be inserted into the dashboard exactly the same as captured from the bar to work.

Figure 1-12: Setting custom Dashboard

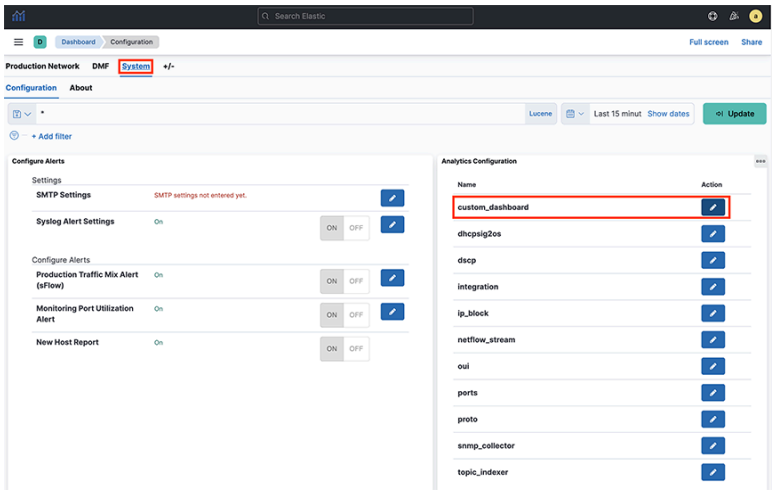
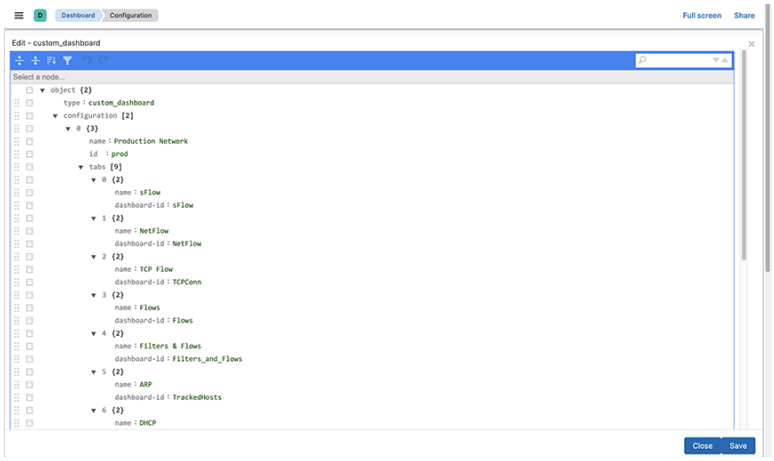
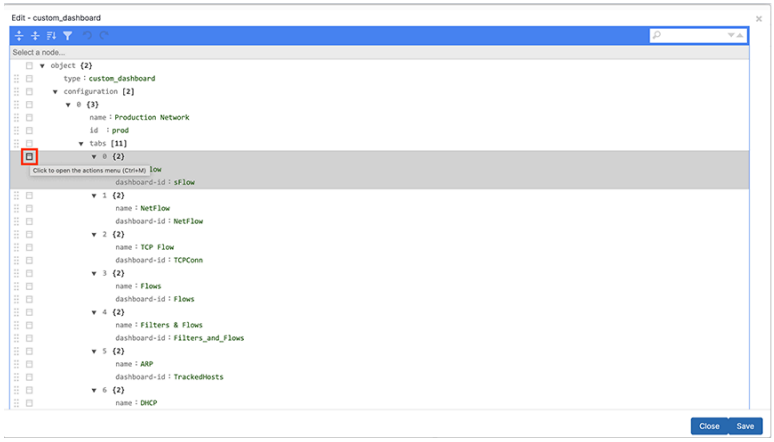


Figure 1-13: Default Dashboard configuration



Open the menu to select the action.

Figure 1-14: Open the action menu



You can select the tab for the duplicate entries.

Figure 1-15: Duplicate the tab

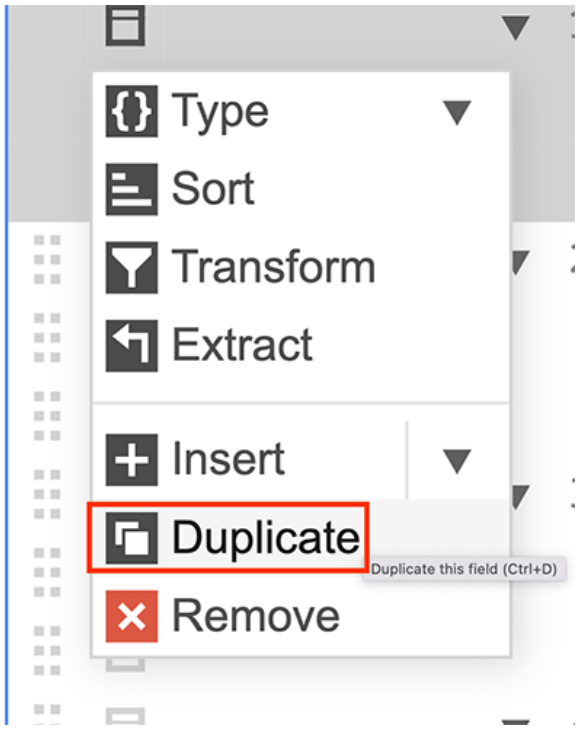
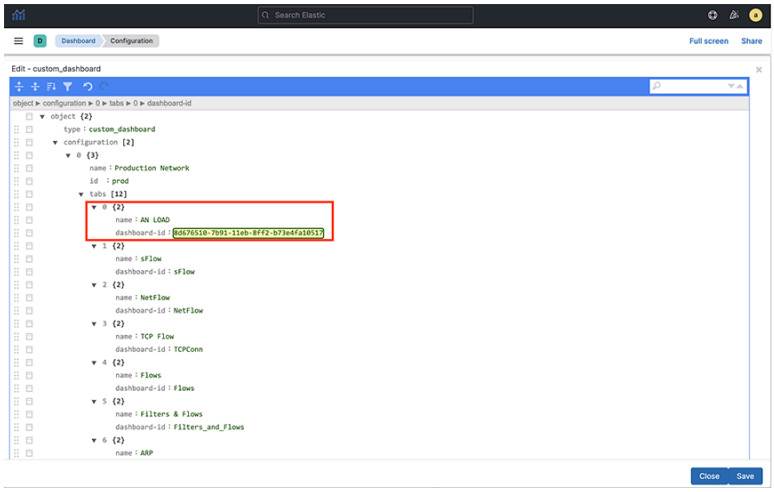
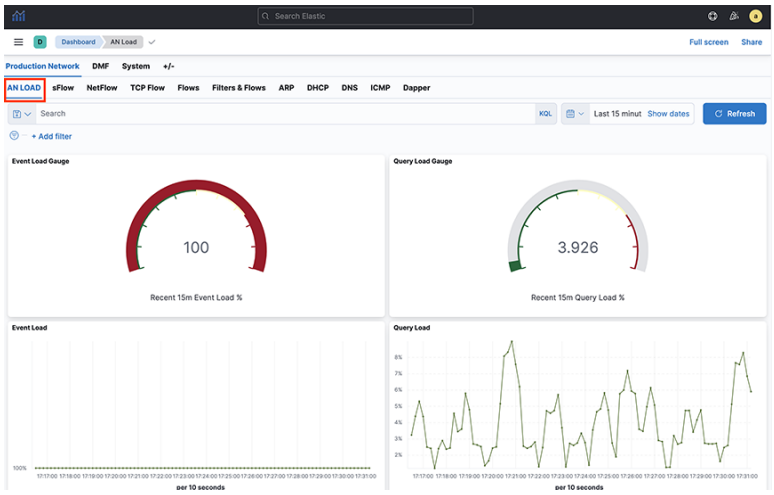


Figure 1-16: Insert the name tag ID



Now the dashboard shows the customization of the option selected by the user.

Figure 1-17: Selected option for the user



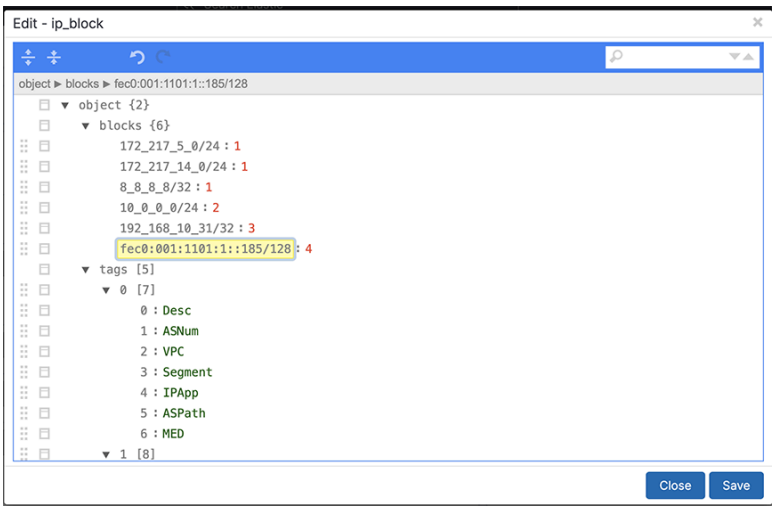
1.7 Mapping IP Address Blocks

You can map an IP address or a range of addresses to a description, which lets you search for description text instead of the IP address. This lets you identify a specific group or organization that is sending or receiving traffic.

To assign a single IP address or a block if IP addresses to a tool, group, or organization, complete the following steps.

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

   Figure 1-18: Edit IP Blocks

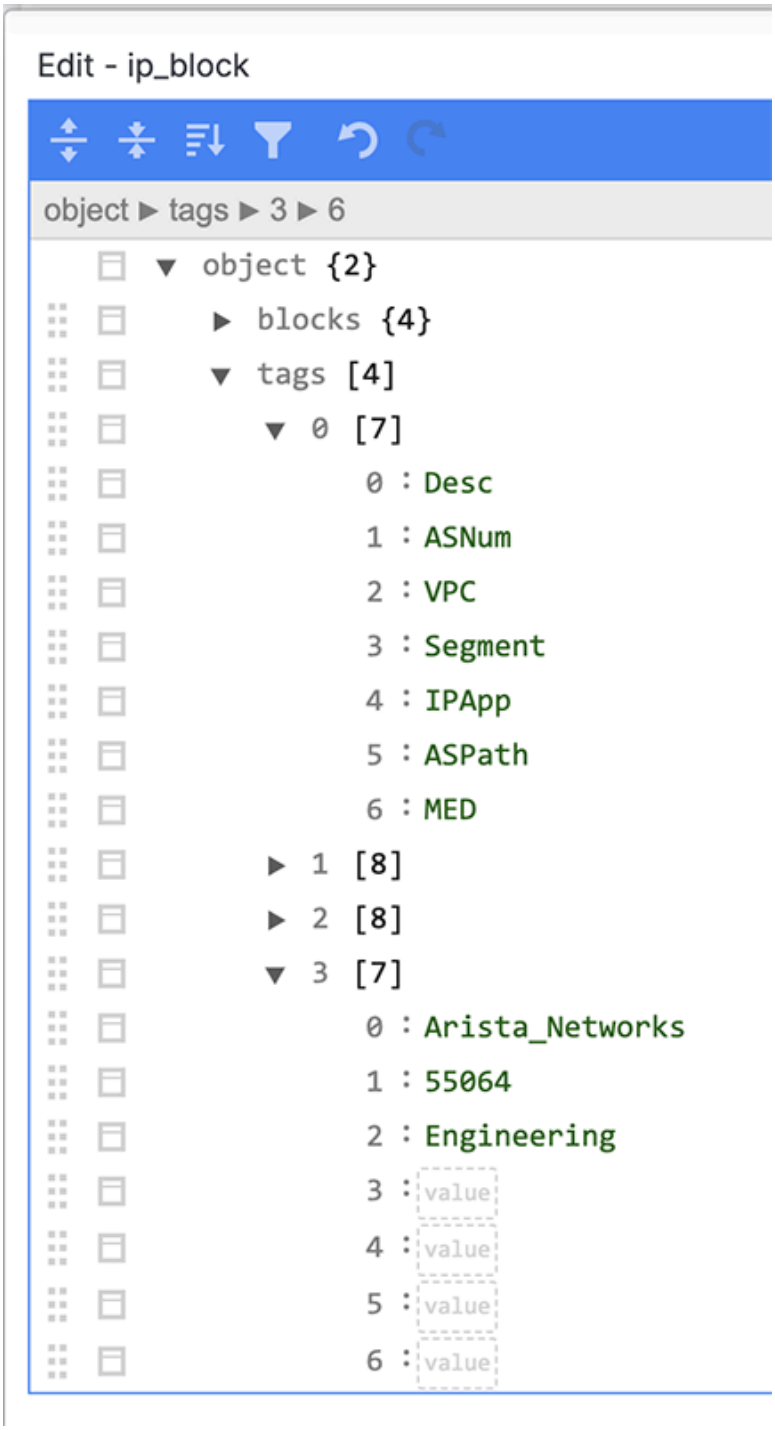


2. Copy an existing block by clicking on any square box along the left and select **Duplicate** from the pop-up menu.

   The duplicated block will be appended to the existing list of blocks and will be 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 1-19: Key Value Pairs



The first four keys are copied automatically. 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), which is 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 enclosed in double quotation marks (“”).
5. (Optional) To define an additional key, select any key and select Duplicate from the pop-up menu. Then type over the existing value with the correct value for the new key.

   The default keys are used in existing dashboards. The added key pairs can be used in customized dashboards. The fifth and sixth keys can be custom keys.

   These keys are added to the flow for the source and destination IPv4 address. For example, source description would be ***sDesc*** and destination description would be ***dDesc***.

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

1.8 Mapping DHCP to OS

The user can map DHCP signatures to known operating systems. These unique signatures are derived from fingerbank.org. As shown in the following image, a number of two-digit numbers are assumed signatures of each OS (derived from ***#ngerbank.org***).

Figure 1-20: Unique OS Signatures from fingerbank.org

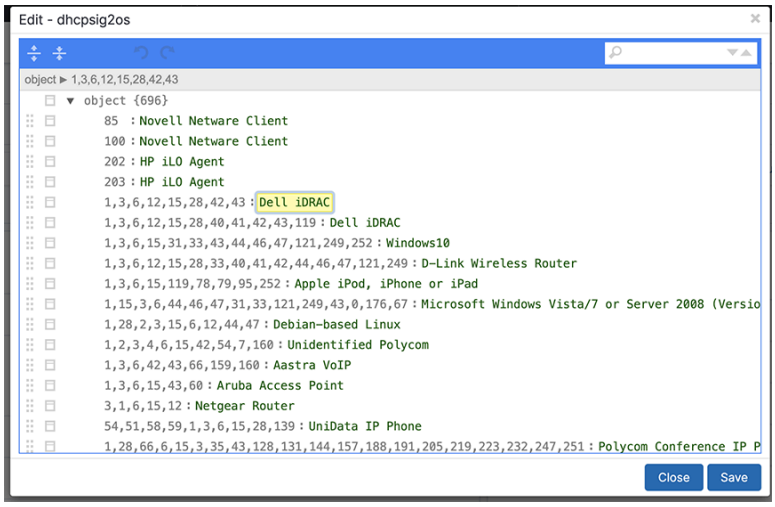
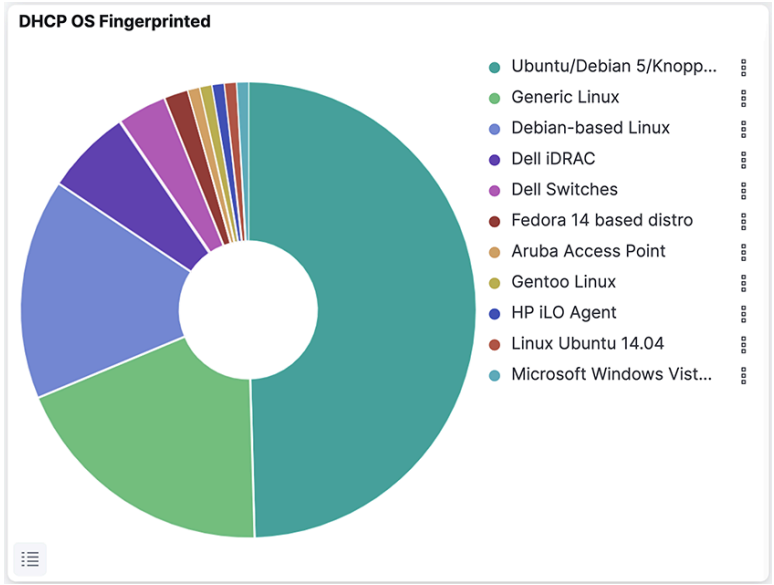


Figure 1-21: OS Information Received through DHCP Signatures



1.9 Mapping Ports and Protocols

The Analytics Node maps typically used ports to their L4 applications. The same is done for protocols. These protocols and ports can also be user-defined for custom application ports and custom protocols.

Figure 1-22: Edit Ports

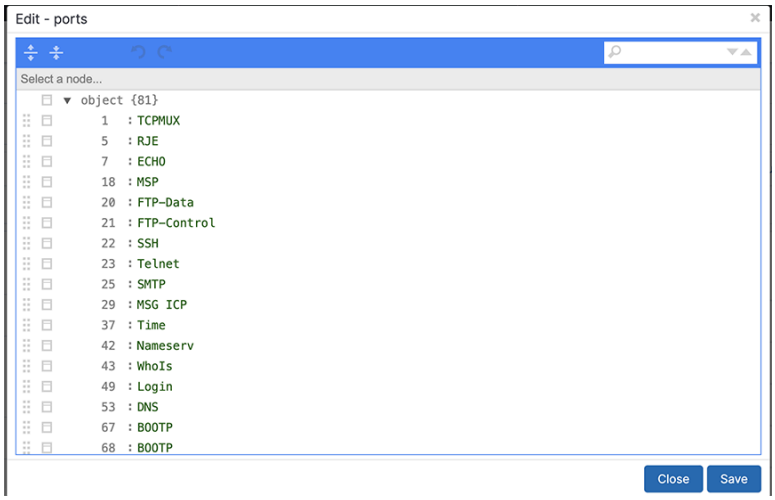
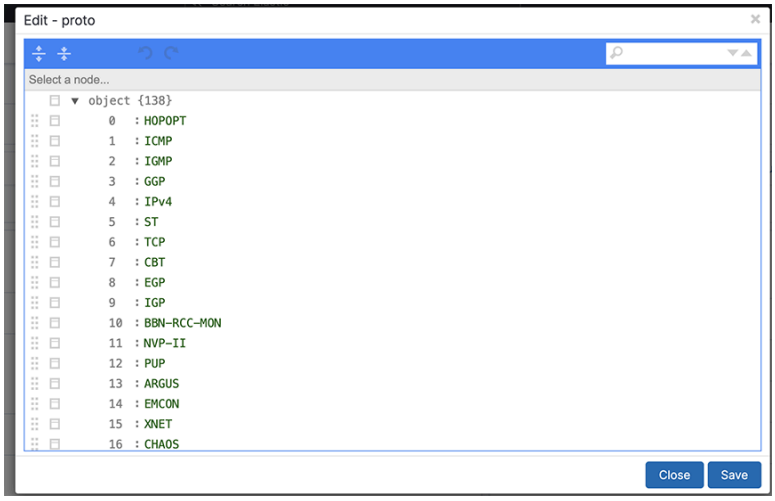


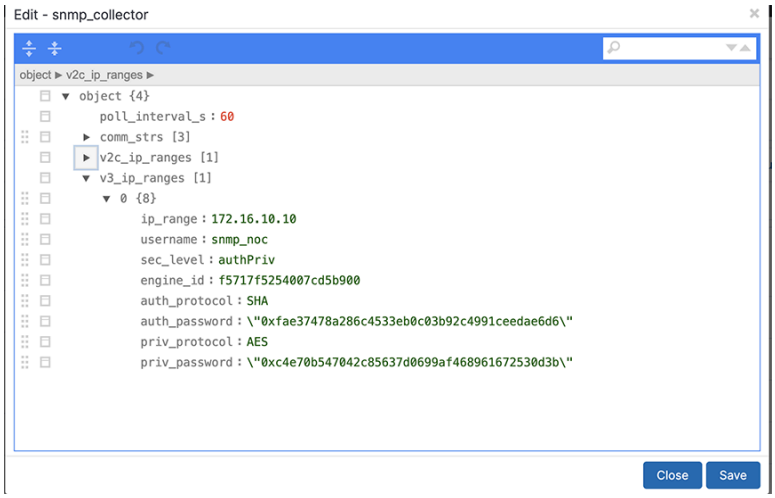
Figure 1-23: Edit Protocols



1.10 SNMP Collector

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

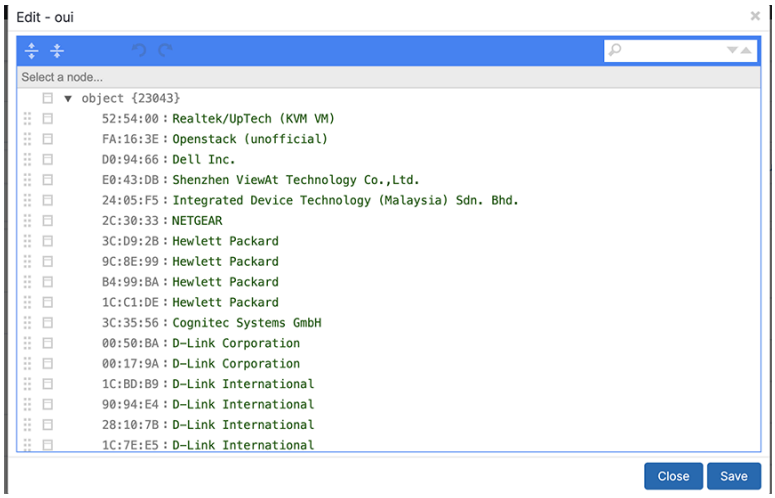
Figure 1-24: SNMP Collector



1.11 Mapping OUI to Hardware

The user can map ARP Organizational Unique Identifiers (OUIs) for various hardware vendors.

Figure 1-25: OUIs of Various Hardware Vendors



1.12 Topic Indexer on Arista Analytics

Description

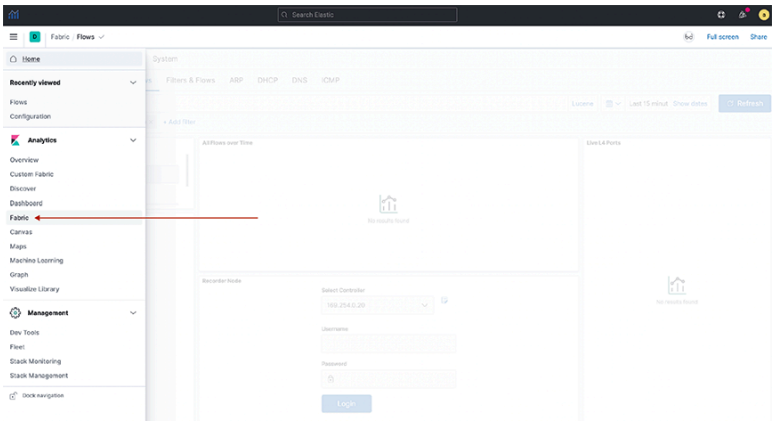
The Analytics Node (AN) incorporates a feature known as topic_indexer, designed to facilitate data ingestion from customer Kafka topics and its subsequent storage into Elasticsearch indices.

This process involves modifying fieldnames and specifying the supported timestamp field during the ingestion phase. The renaming of fieldnames 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.

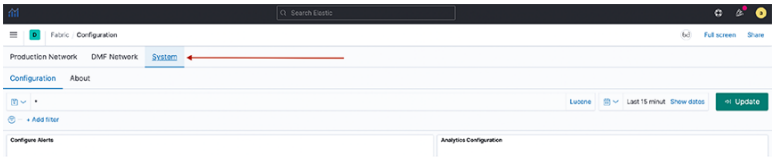
Implementation Details

- 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 1-26: Analytics > Fabric**

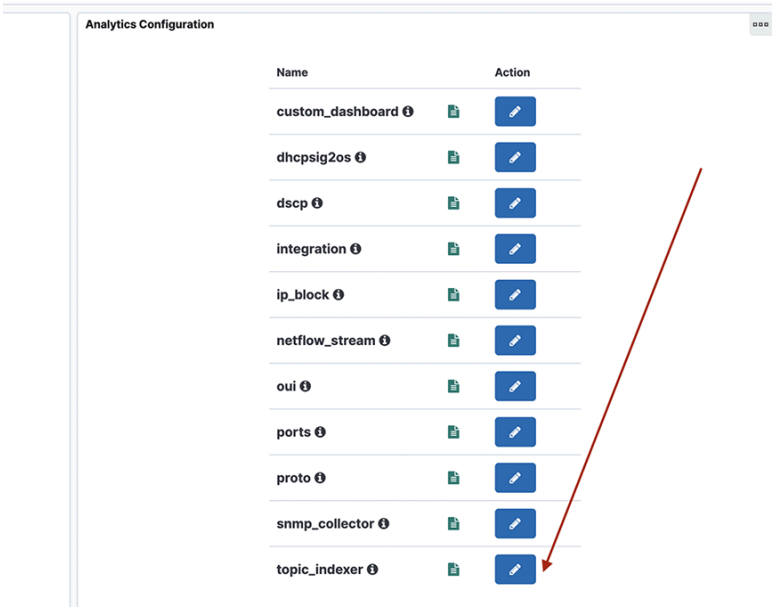


- Another view:

Figure 1-27: System > Analytics Configuration



- The configuration for a topic to be consumed can be specified as described in the design section.
- **Figure 1-28: Node selection**

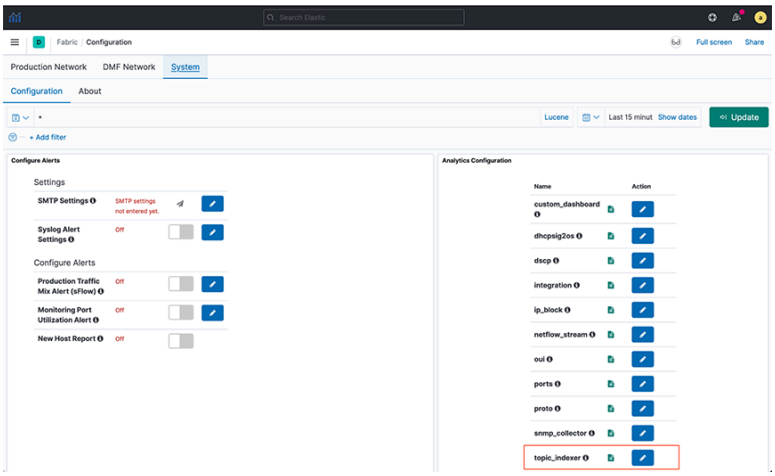


Configuration

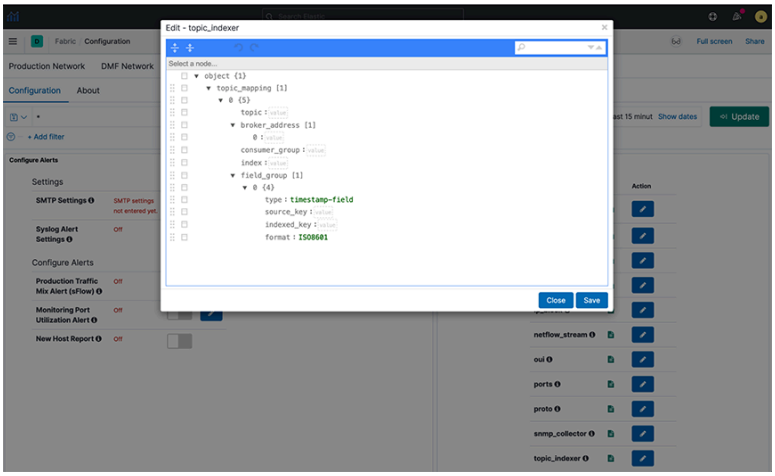
Kibana Configuration

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

  Figure 1-29: System > Configuration



- **Figure 1-30: Topic indexer configuration**



Field Details

Each topic is mapped 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. We create it as topic_name + index_name. Setting this field will be particularly useful when ingesting multi-partitioned topics from the client's end.
- **index**: A dedicated index name we want for the topic; type string. In Elastic Search (ES), it will be 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.

  Specifies format for modifications to incoming data.
- **type**: This is to set timestamp-field as the type.
- **source_key**: This is the source field name in the incoming data.
- **indexed_key**: This is the destination field name inserted in the outgoing ES index.

The indexed_key may be a @timestamp field of an ES index. If we do not specify a @timestamp field, topic_indexer automatically picks the time the message was received 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 an invalid field text.

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

Some 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.
- While entering the broker address, use the format: IPv4:PORT.

Application Scenario

Querying Across DataStream using runtime-fields

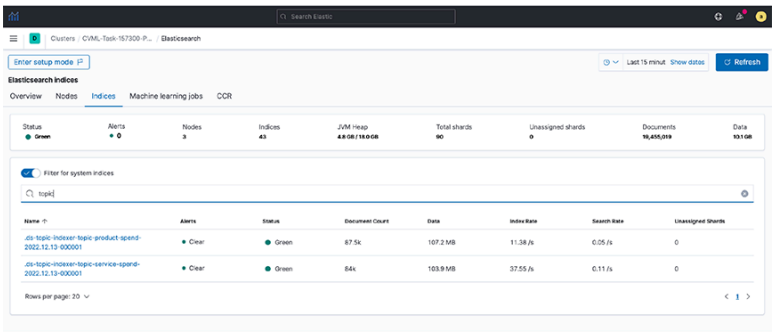
Use runtime fields when making complex changes beyond simply renaming a field, such as converting it from a string type to an IP address. After every change to a runtime field, issue a

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

Note: These changes are not persistent. You must reapply them after any restart of AN.

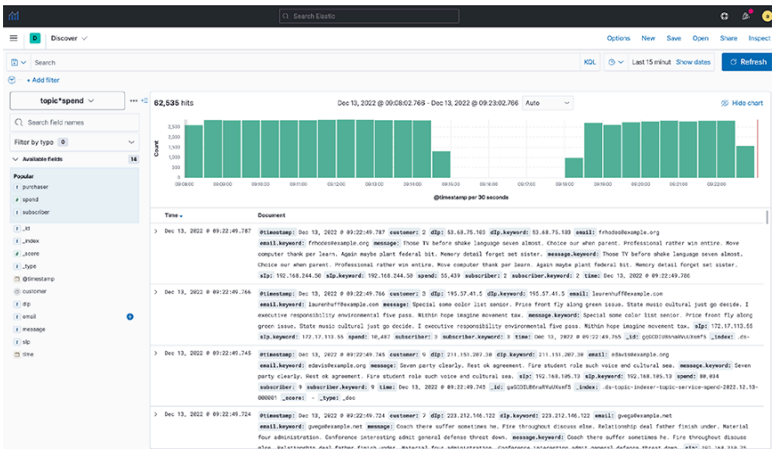
Use Case:

- Cross-index visualization - two data streams that need cross-querying:
- **Figure 1-31: Cross index visualization**



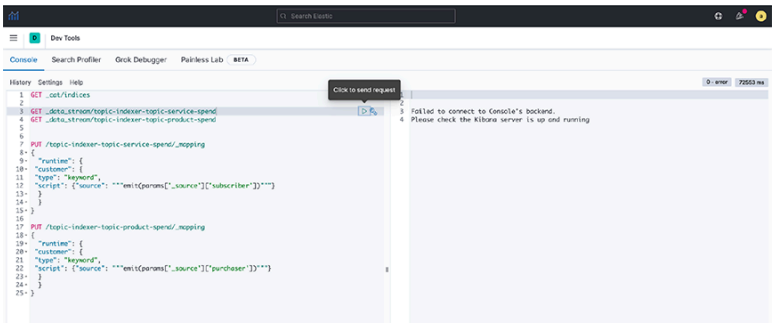
- **Step 1.** To view the documents in these indexes, create an index pattern (e.g., topic*spend) in Kibana.
- **Step 2.** View the data in the Discover dashboard.

Figure 1-32: Discover dashboard



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

Figure 1-33: Dev Tools



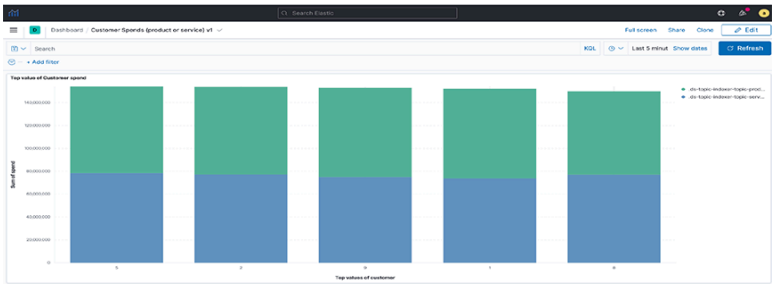
Note: Setting rollover policy on runtime fields can also be done in Dev Tools, as shown in the following examples:

```
POST /topic-indexer-service-spend/_rollover
POST /topic-indexer-product-spend/_rollover
```

Note: These changes are not persistent. You must reapply them after any restart of AN.

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

  Figure 1-34: 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
```

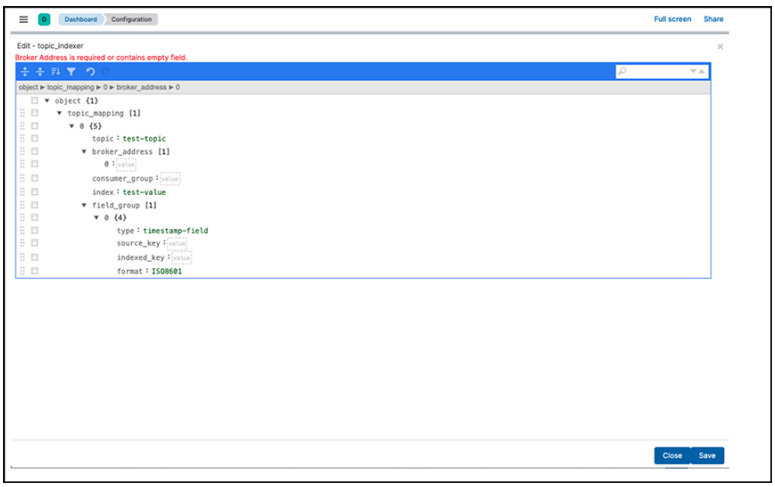
Troubleshooting

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

1. **The Save button in 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 when filling in the config for topic indexer as stated in the above section linked: Topic Indexer on Arista Analytics . As illustrated below, you may encounter validation errors when entering an invalid value in the configuration field. Topic Indexer on Arista Analytics

   Figure 1-35: 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. **Index for topic indexer is not 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 name of the index is prefixed by topic_indexer_.

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 once again to see whether the spellings for 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 in the configuration. For more details, please review the following links:
    1. Kafka 3.5 Documentation
    2. Kafka Listeners – Explained | Confluent
c. If all the above steps are correct, you should now check the logs in the Analytics Node for the topic indexer.

    Steps to reach the topic_indexer.log file in 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 into 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 above), use the topic_indexer log file to validate the problem.

5. **Index pattern for topic indexer is missing.**

In the Kibana UI, we create 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:

Create an index pattern | Kibana Guide [7.17] | Elastic

Chapter 2

Production Network Monitoring

This chapter describes the dashboards provided on the Production Network tab, in which you can view traffic and events that occur on the production network interfaces connected to the DANZ Monitoring Fabric. This chapter includes the following sections:

2.1 sFlow

The sFlow dashboard is displayed by default when you click the **Fabric** option. 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
- Source Port
- Top Destinations
- Destination Port
- Traffic over time
- Flow by Filter Interface
- Flow by Device & IF
- Count sFlow vs. Last Wk
- Flow QoS PHB
- Flow Source
- Flow Destination
- sFlow MTU Distribution
- Flows by Time

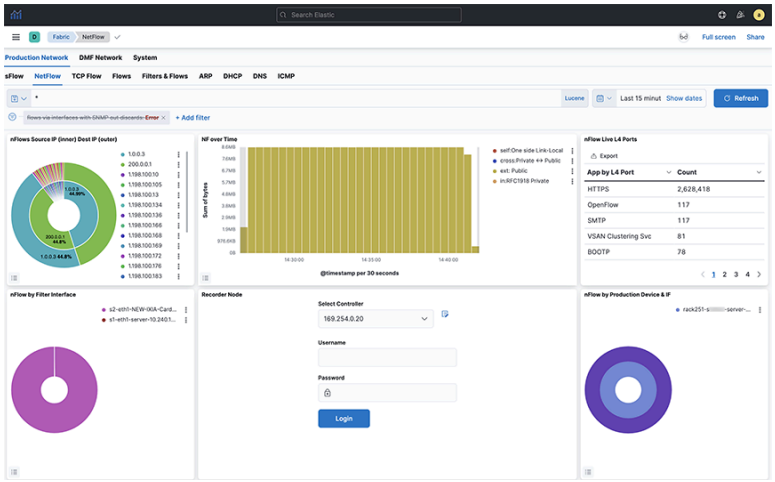
2.1.1 sFlow and VXLAN

The sFlow dashboard shows both outer and inner flows of VXLAN packets based on the VNI number of the VXLAN packet. To see all the inner flows of a particular VXLAN packet, first filter by VXLAN packets on the **App L4 Port** window to display all of the VXLAN packets. Identify the VXLAN packet you are interested in from the **Flows by Time** window. Expand the row, and note the VNI number of the packet, then remove the VXLAN filter and filter based on the VNI number. This will show both the outer flow of the VXLAN packet and all the inner flows associated with that VXLAN packet.

2.2 NetFlow and IPFIX

When you click **NetFlow**, the system displays the following dashboard:

Figure 2-1: Production Network > NetFlow Dashboard



To obtain NetFlow packets, you must configure the NetFlow collector interface on the Arista Analytics Node, as described in the Setting up the NetFlow Collector on the Analytics Node section.

The NetFlow dashboard summarizes information from the NetFlow messages sent to the Arista Analytics Node from the DANZ Monitoring Fabric controller or other NetFlow flow exporter, and provides the following panels:

- nFlow Source IP (inner) Destination IP (outer)
- NF over Time
- nFlow Live L4 Ports
- nFlow by Filter Interface
- nFlow by Production Device & IF
- NF by QoS PHB
- NF by DPI App Name
- NF Top Talkers by Flow
- NF Detail

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.

Starting from ***BMF-7.2.1*** release, the Arista Analytics Node can also handle NetFlow V5/V9 and IPFIX traffic. All of the flows represent with a Netflow index. From the NetFlow Dashboard, filter rules apply to display specific flow information.

Figure 2-2: NetFlow Version 5

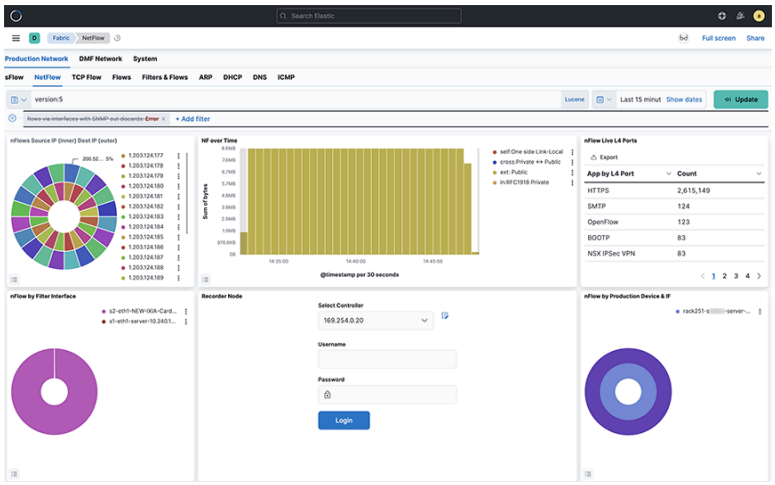


Figure 2-3: NetFlow Version 9

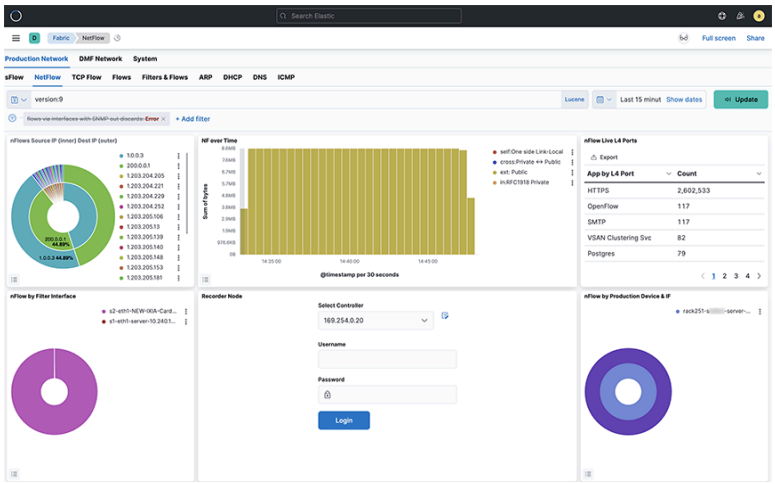
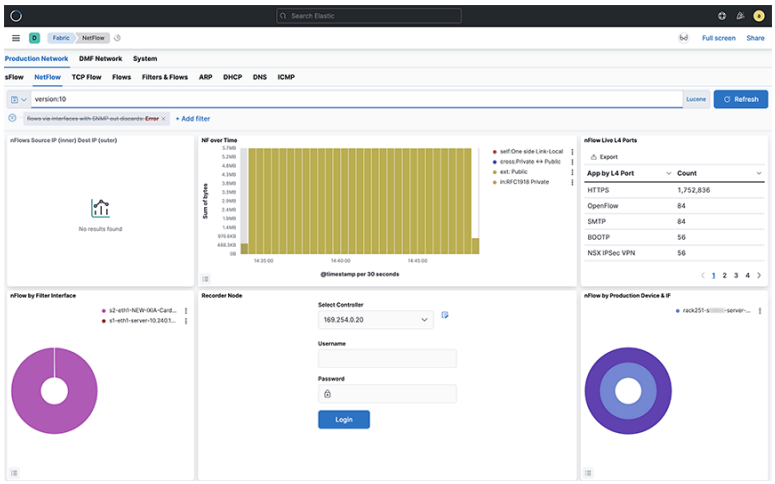


Figure 2-4: NetFlow Version 10



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. Starting from the ***DMF-8.1.0*** release, Analytics Node capability augment in support of the following Arista Enterprise-Specific Information Element IDs:
   - 1036 -AristaBscanExportReason
   - 1038 -AristaBscanTsFlowStart
   - 1039 -AristaBscanTsFlowEnd
   - 1040 -AristaBscanTsNewLearn
   - 1042 -AristaBscanTagControl
   - 1043 -AristaBscanFlowGroupId

2.2.1 Consolidating Netflow V9/IPFIX records

The user can consolidate Netflow V9 and IPFIX records by grouping those sharing similar identifying characteristics within a configurable time window.

This reduces the number of documents published in ElasticSearch, reducing disk usage and increasing efficiency, specially for long flows where a 40:1 consolidation has been observed.

In case of low flow rate for packets, it is recommended to not enable this consolidation. It may result in delay in the publication of documents.

The following configuration sets the load-balancing policy of Netflow/IPFIX traffic among nodes in a 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. It is recommended to use this, since consolidation operations are performed on each node independently.
- **Round-robin**: distributes the packets equally between the nodes, if source-hashing results in traffic distribution being significantly unbalanced. Round-robin is the default behavior.

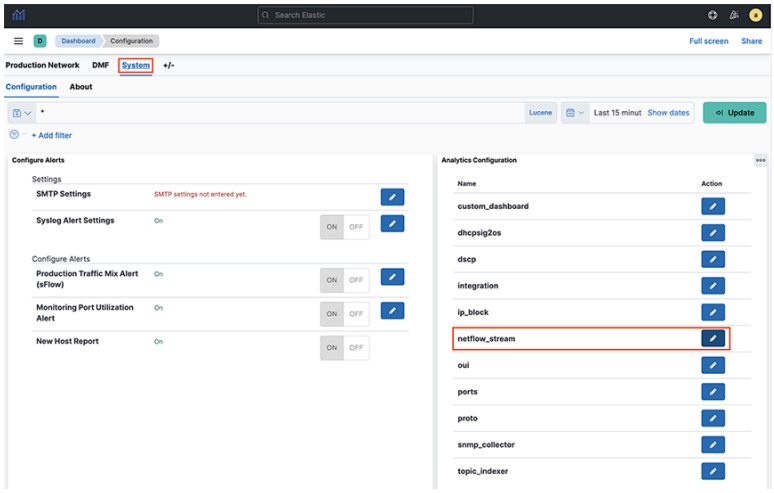
Note: It is recommended to configure the round-robin to lighten the load on the leader node, when flow rate is higher than 10k/sec in cluster setup.

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

Kibana Setup

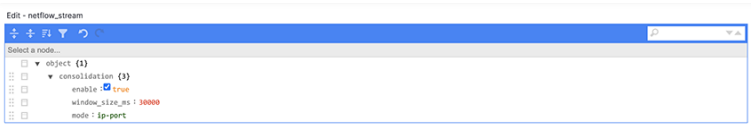
To perform the Kibana configuration, select the **System** > **Configuration** tab on the Fabric page and open the **Analytics Configuration** > **netflow_stream** panel:

Figure 2-5: Kibana setup



For editing the netflow stream, go to the following tab:

Figure 2-6: Edit the netflow stream



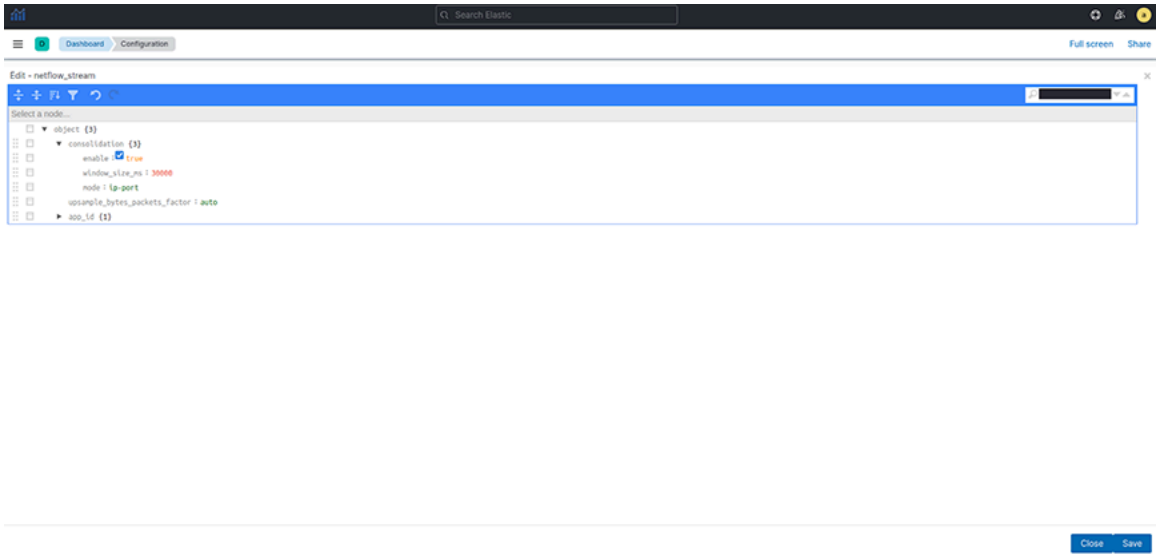
There are three required settings:

- **enable**: enables or disables the consolidation.
- **window_size_ms**: window size is adjusted by the rate of Netflow V9/IPFIX packets per second received by the analytics node. By default, the window is set to 30 seconds, though it is measured in millisecond.
- **mode**: There are three supported modes:
  - **ip-port**: consolidates records with the same source IP address, destination IP address, IP protocol number, and lower numerical value of source or destination Layer 4 port number.
  - **dmf-ip-port-switch**: consolidates records from common DMF Filter switches that also meet **"ip-port"** criteria.
  - **src-dst-mac**: consolidates records with the same source and destination MAC address.

    Note: Only use this mode with Netflow V9/IPFIX templates collecting Layer 2 fields.

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

Figure 2-7: 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.

2.2.2 NetFlow and IPFIX Flow with Application Information

This section describes a new feature of Arista Analytics that combines Netflow and IPFIX records containing application information with Netflow and IPFIX records containing flow information.

This feature improves the visibility of data per application by correlating flow records with applications identified by the flow exporter.

This release supports only applications exported from Arista Networks Service Nodes. In a multi-node cluster, load balancing must be configured 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.

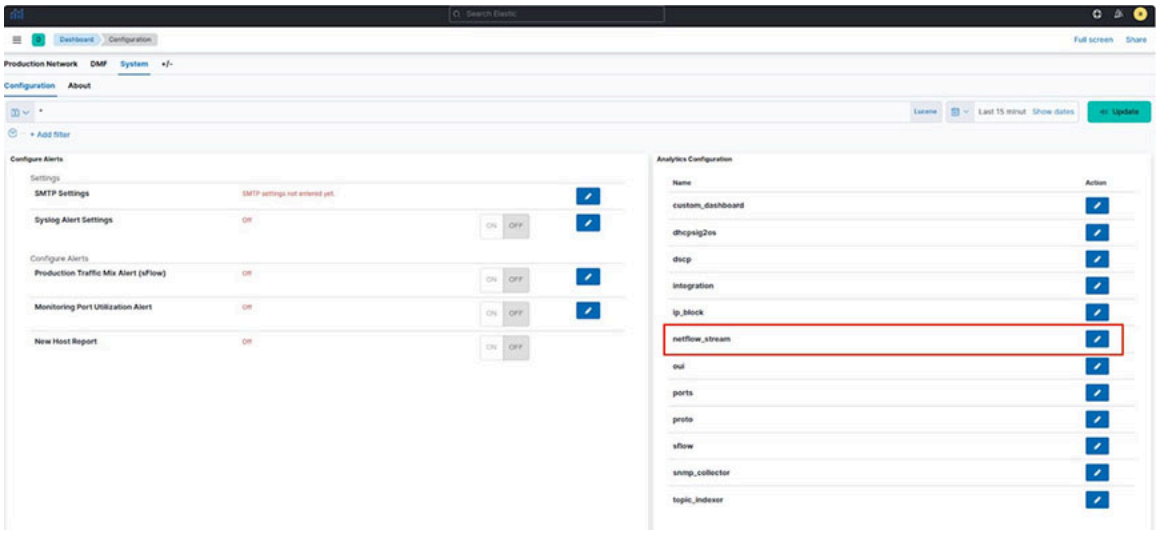
```
analytics# config
analytics(config)# analytics-service netflow-v9-ipfix
analytics(config-an-service)# load-balancing policy source-hashing
```

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

Kibana Configuration

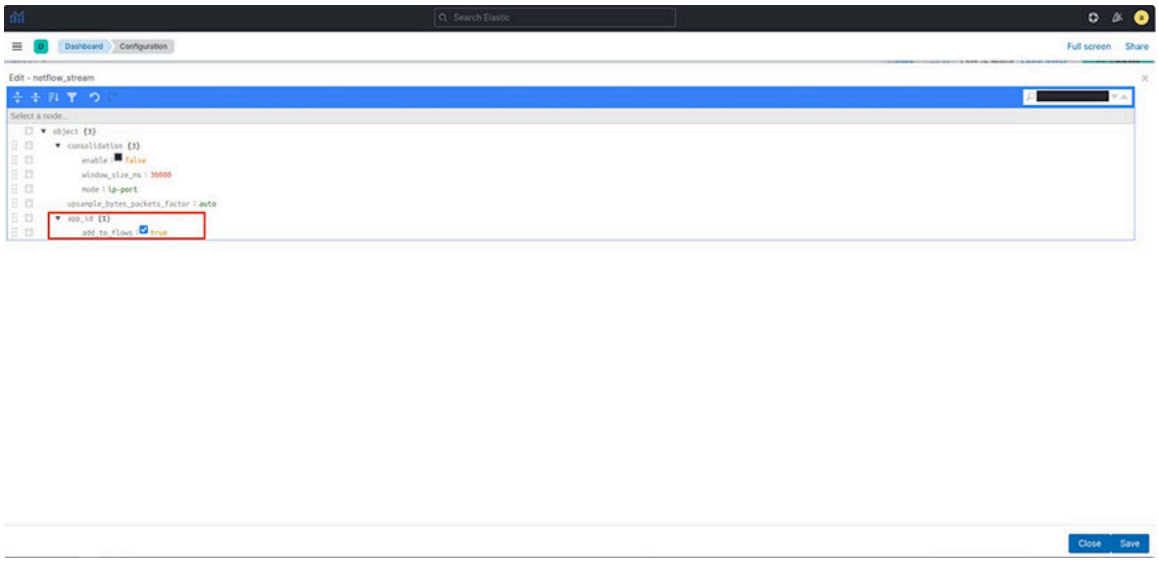
To perform the Kibana configuration, select the **System** > **Configuration** tab on the Fabric page and open the **Analytics Configuration** > **netflow_stream** visualization.

Figure 2-8: Dashboard - Netflow stream configuration



Add the **app_id** configuration object.

Figure 2-9: Edit - Netflow stream



In the **app_id** configuration object, one setting is required:

- **add_to_flows**: Enables or disables 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 only if the application record is received 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.

2.2.3 NetFlow and sFlow Traffic Volume Upsampling

This feature of Arista Analytics offers the ability to upsample traffic volume sampled by NetFlow V9/IPFIX and sFlow. This feature provides a 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 provides 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 ***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 2-10: Dashboard - Netflow IPFIX configuration

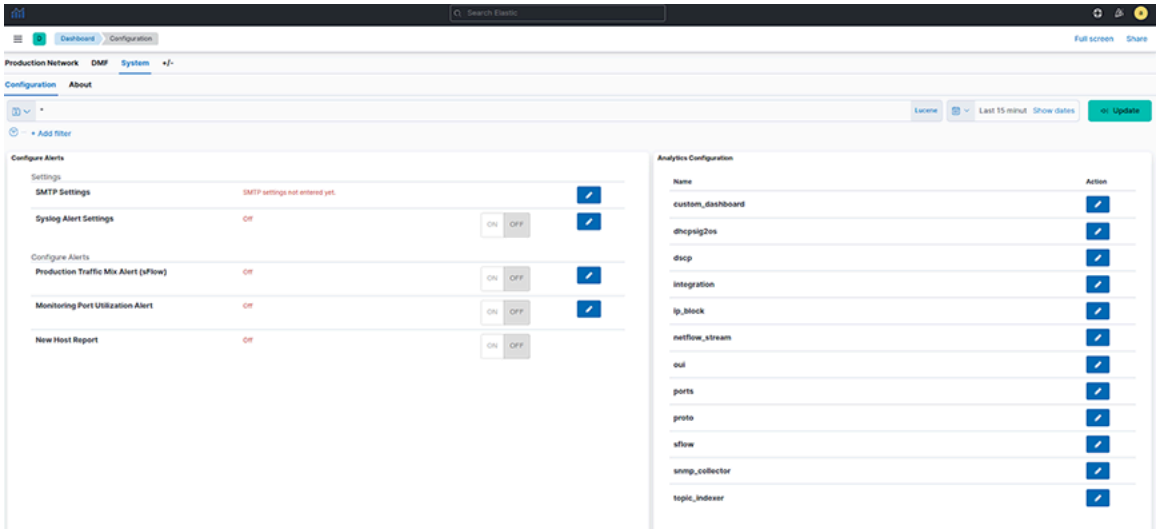
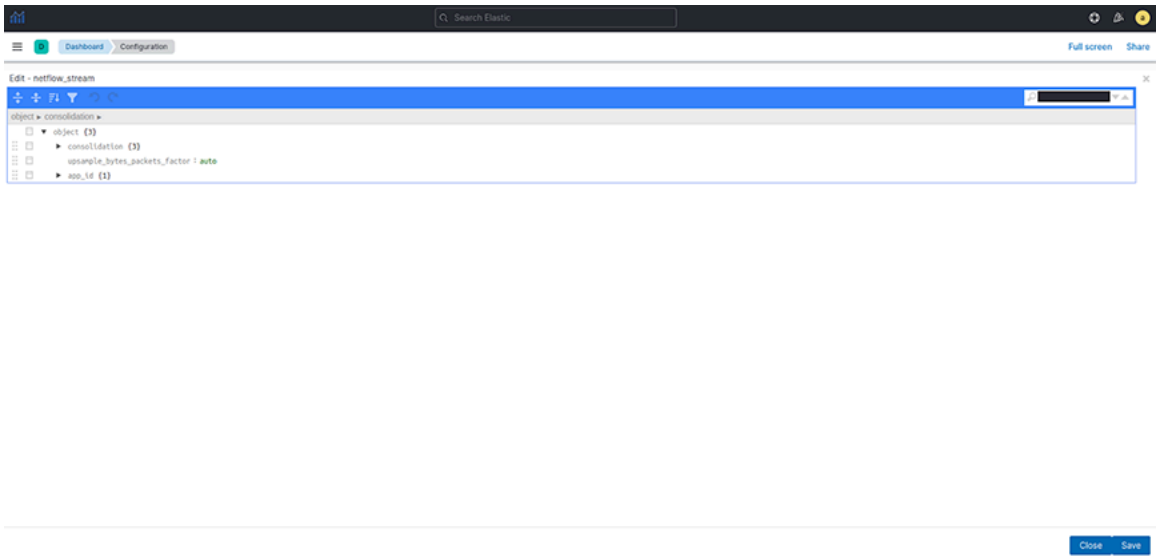


Figure 2-11: Edit - Netflow IPFIX



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

- **Auto**: This is the default option. ***DMF 8.5.0*** does not support automated upsampling for Netflow V9/IPFIX. Arista Networks recommends configuring an integer if upsampling is needed.
- **Integer**: Multiply the number of bytes and packets for each collected sample by 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 2-12: Dashboard - sFlow configuration

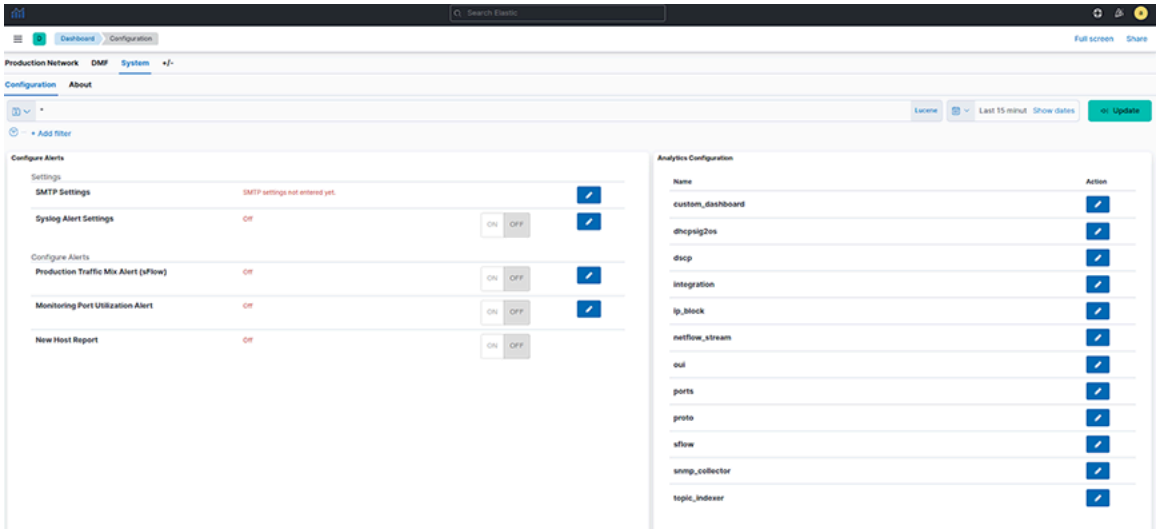
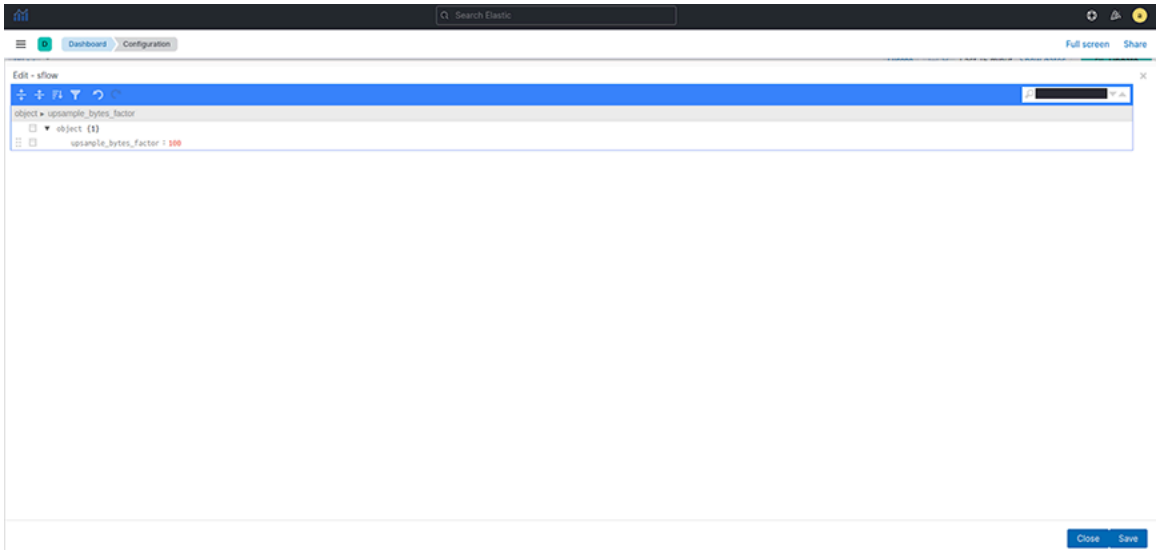


Figure 2-13: Edit - sFlow



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

- **Auto**: Approximate the number of 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 by this configured number.

Dashboards

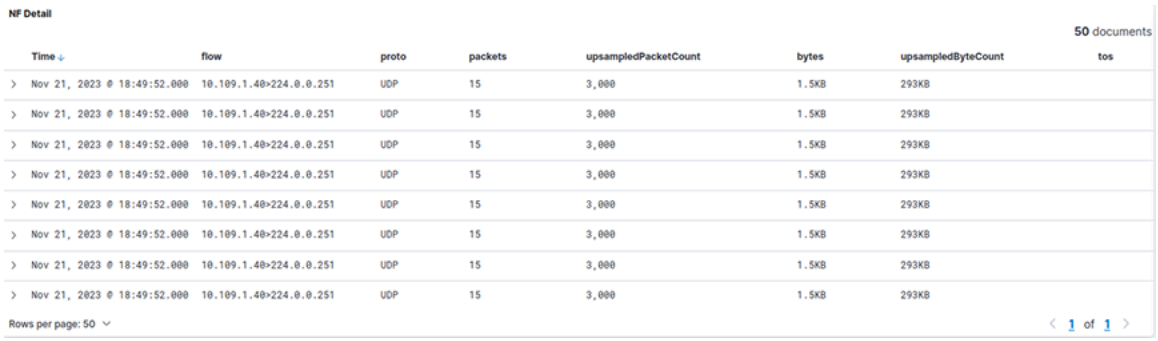
NetFlow Dashboard

The NetFlow dashboard is on the **Production Network** > **NetFlow** tab on the Fabric page. The following visualization will display upsampled statistics:

- **NF over Time**

- **NF Top Talkers by Flow**

Figure 2-14: NF Detail visualization



The ***DMF 8.5.0*** release adds two new columns:

- **upsampledPacketCount**: Approximate total count of packets for a flow.
- **upsampledByteCount**: Approximate total count of bytes for a flow.

Note: In ***DMF 8.5.0***, configuring upsampling to **Auto**, **upsampledByteCount** and **upsampledPacketCount** will copy the **bytes** and **packets** column and display the values of **bytes** and **packets** in the graphs and tables of this dashboard.

sFlow Dashboard

The SFlow dashboard is on the **Production Network** > **sFlow** tab on the Fabric page. The **Traffic over Time** visualization will display upsampled statistics.

Figure 2-15: Flow by Time visualization

The newly added **upsampledByteCount** represents a flow's approximate total count of bytes.

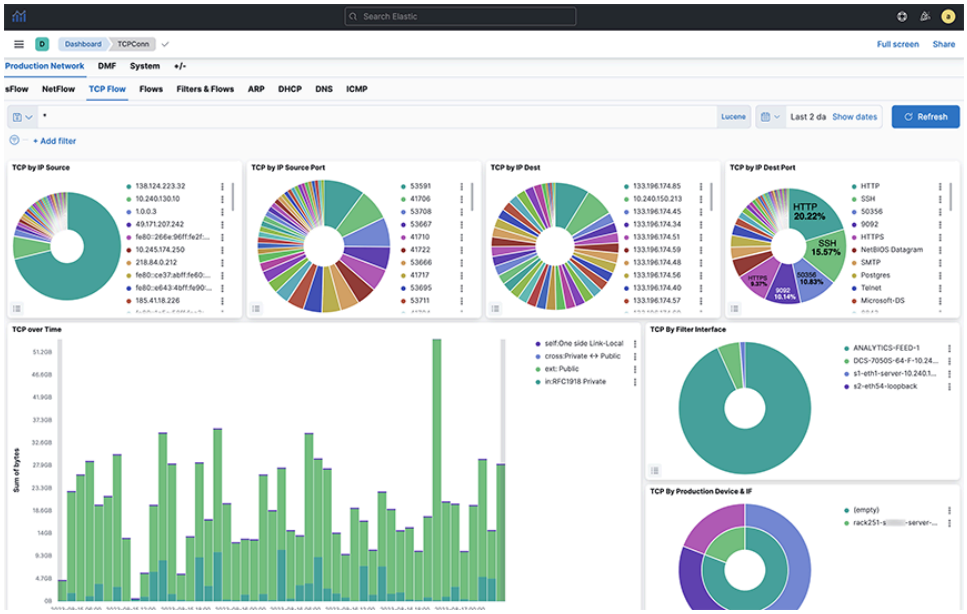
Troubleshooting

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

2.3 TCPFlow

When you click the **TCPFlow** tab, the system displays the following dashboard.

Figure 2-16: Production Network > TCPFlow Dashboard

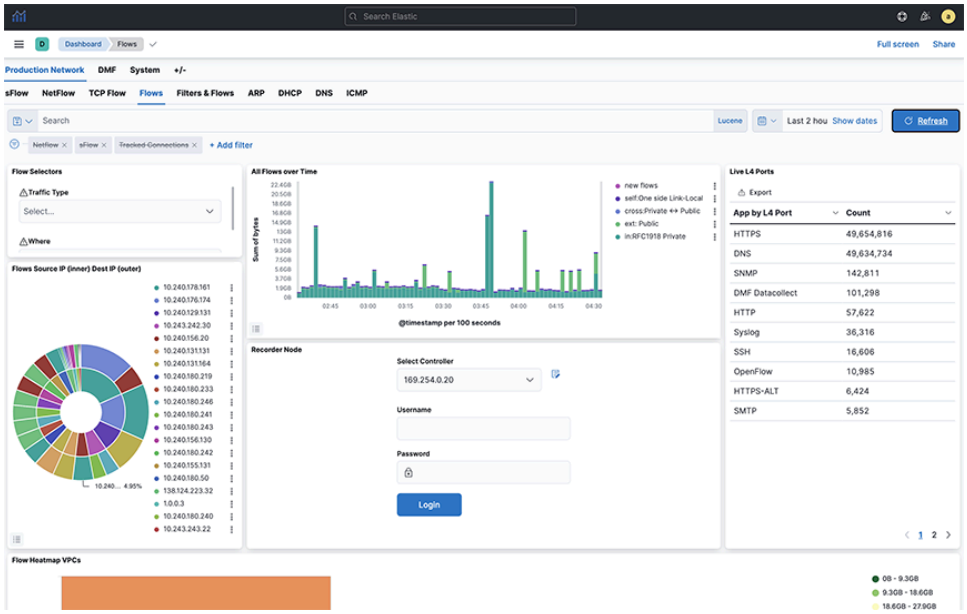


The information on the TCPFlow dashboard depends on TCP handshake signals and deduplicates. The **Filter Interface** visualization indicates the filter switch port where data is received. The switch description is specified in the Description attribute of each switch, configured on the DANZ Monitoring Fabric controller. **Device & IF** on this dashboard refers to the end device and depends on LLDP packets received.

2.4 Flows

When you click the **Flows** tab, the system displays the following dashboard.

Figure 2-17: Production Network > Flows Dashboard



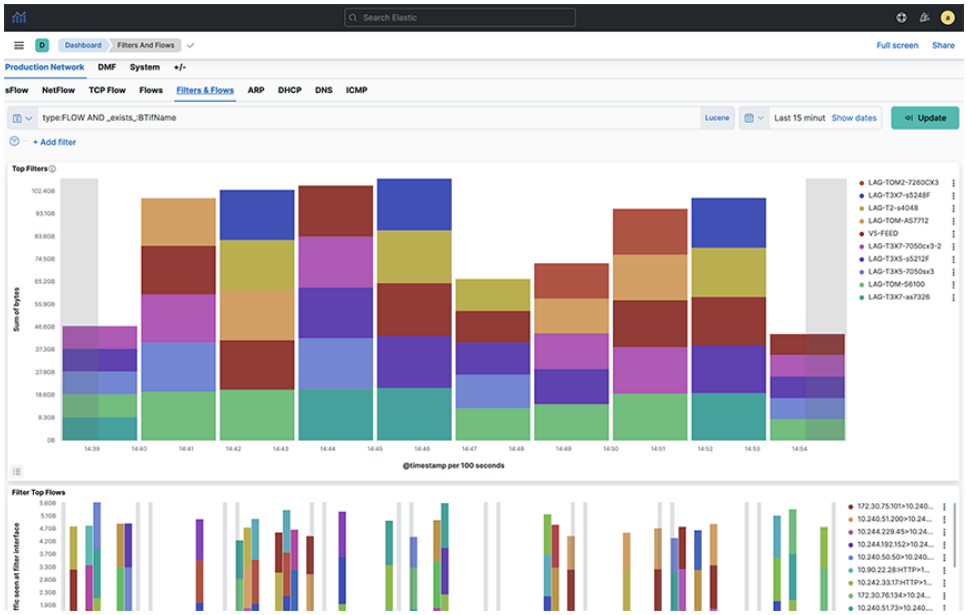
The **Flows Dashboard** summarizes information from sFlow and NetFlow messages and provides the following panels:

- All Flows Type
- All Flows Overtime
- All Flows Details

2.5 Filters & Flows

When you click the **Filters & Flows** tab, the system displays the following dashboard.

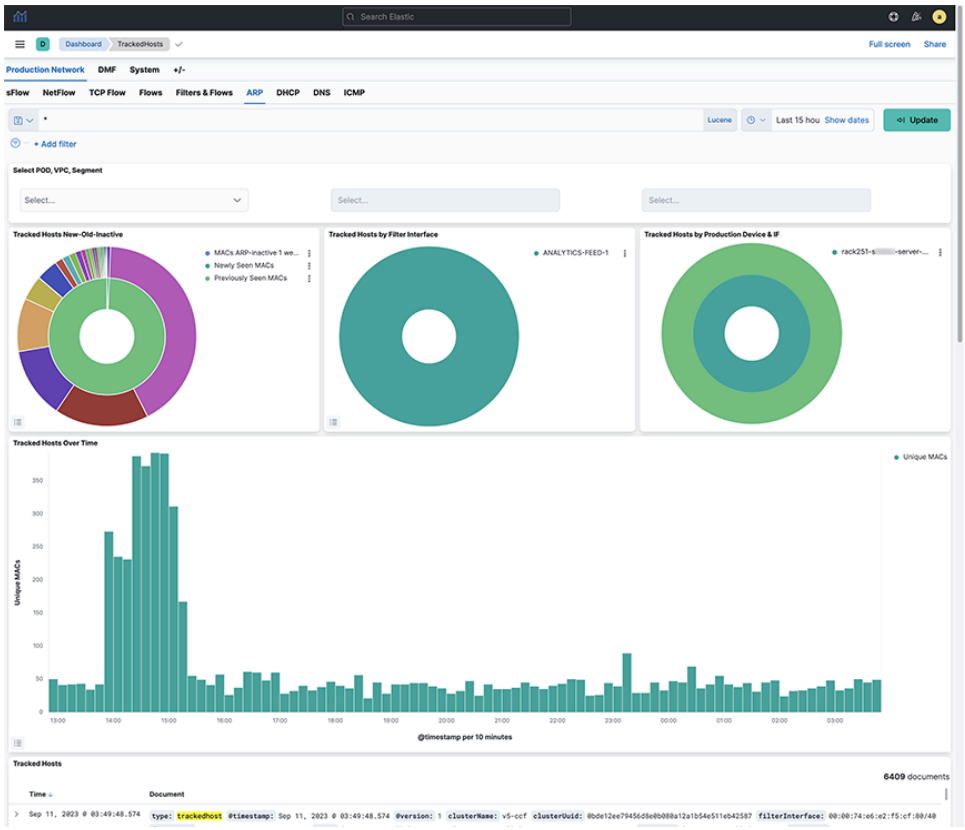
Figure 2-18: Production Network > Filters & Flows Dashboard



2.6 ARP

When you click the **ARP** tab, the system displays the following dashboard. This data correlates by the tracked host feature on the DANZ Monitoring Fabric controller. You see All ARP data over time when you switch interface and production devices.

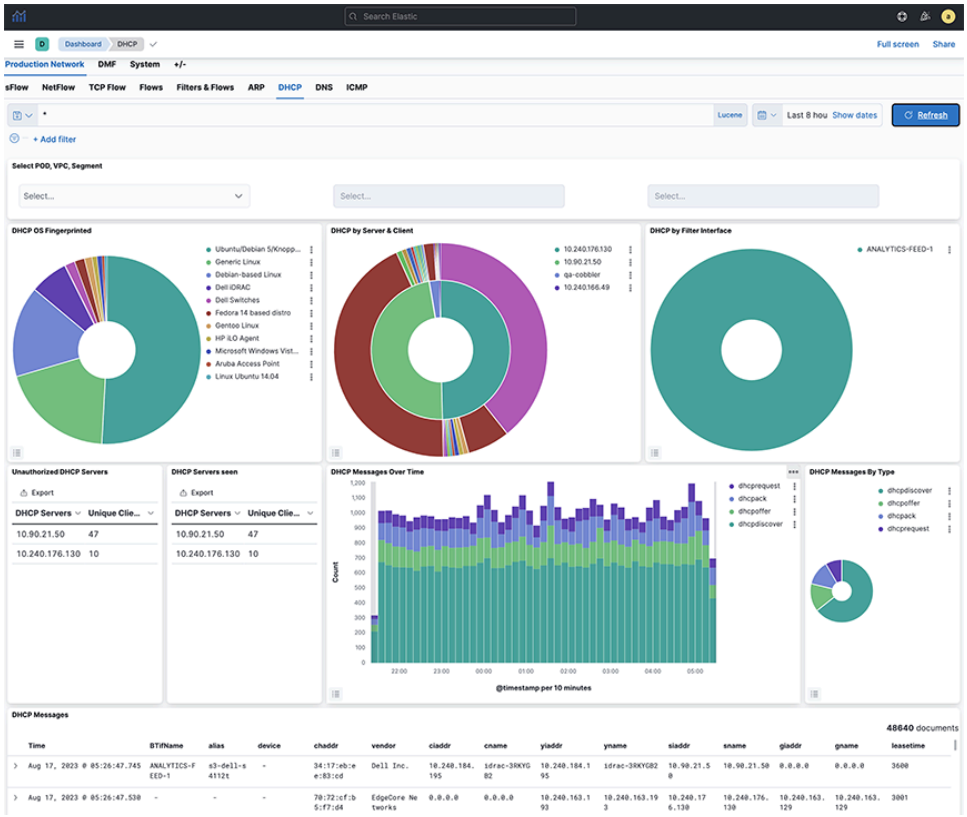
Figure 2-19: Production Network > ARP Dashboard



2.7 DHCP

When you click the **DHCP** tab, the system displays the following dashboard.

Figure 2-20: Production Network > DHCP Dashboard



Note: You can see information about operating systems on the network and data by filter interface and production device.

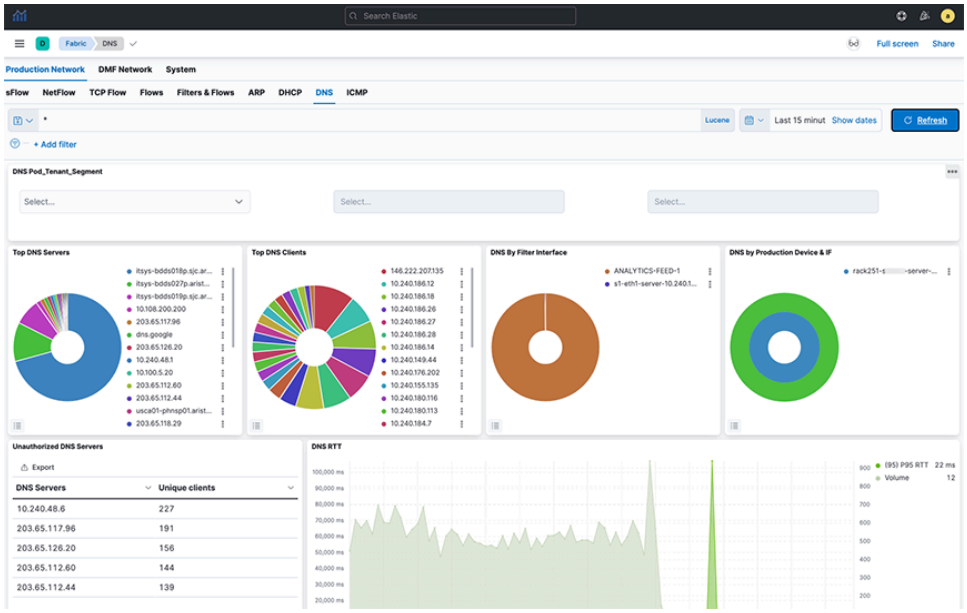
The **DHCP Dashboard** summarizes information from analyzing DHCP activity and provides the following panels:

- DHCP OS Fingerprinted
- DHCP Messages by Filter Interface
- DHCP Messages by Production Switch
- Non-whitelist DHCP Servers
- DHCP Messages Over Time
- DHCP Messages by Type
- DHCP Messages

2.8 DNS

When you click the **DNS** tab, the system displays the following dashboard.

Figure 2-21: Production Network > DNS Dashboard



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

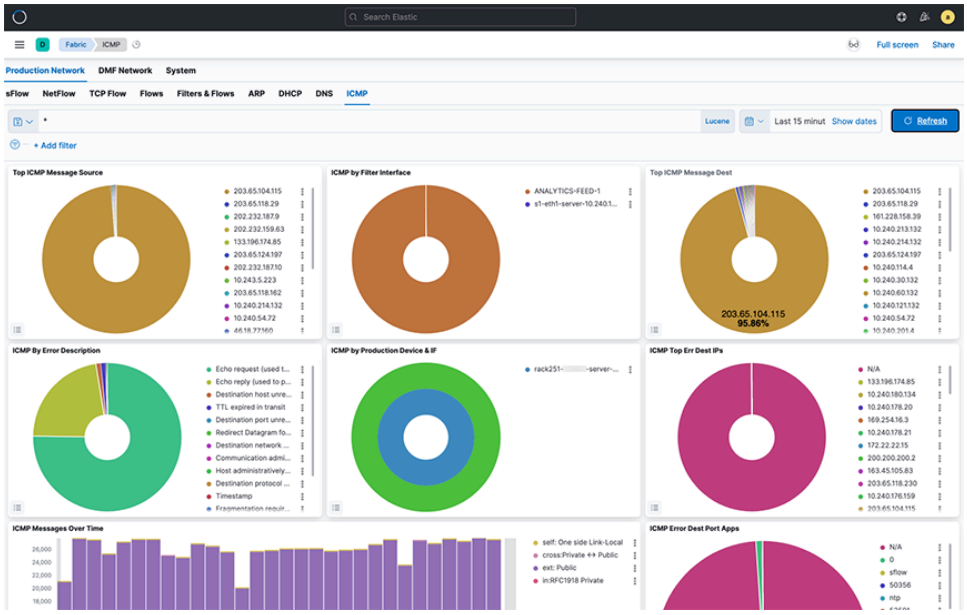
- DNS Top Servers
- DNS Top Clients
- DNS By Filter Interface
- DNS by Production Device & IF
- DNS Messages Over Time
- Unauthorized DNS Servers
- DNS RTT
- DNS All Messages
- DNS RCode Distro
- DNS QType Description
- DNS Top QNames

Note: The DNS RTT value computes using the query and response packet timestamps. If a query packet does not answer by a response packet within ***180*** seconds, the RTT value sets to ***-1***.

2.9 ICMP

When you click the **ICMP** tab, the system displays the following dashboard.

Figure 2-22: Production Network > ICMP Dashboard



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

- Top ICMP Message Source
- ICMP by Filter Interface
- Top ICM Message Dest
- ICMP by Error Description
- ICMP by Production Switch
- ICMP Top Err Dest IPs
- ICMP Top Err Dest Port Apps
- ICMP Messages Over Time
- ICMP Table

Chapter 3

Using the DMF Recorder Node with Analytics

This chapter describes Arista Analytics to use with the DMF Recorder Node. It includes the following sections.

3.1 Overview

The DMF Recorder Node records packets from the network to disk and recalls specific from disk quickly, efficiently, and at scale. A single DANZ Monitoring Fabric controller can manage multiple DMF Recorder Nodes, delivering packets for recording through DANZ Monitoring Fabric policies. The controller also provides central APIs for interacting with DMF Recorder Nodes to perform packet queries across one or multiple recorders and for viewing errors, warnings, statistics, and the status of connected recorder nodes.

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

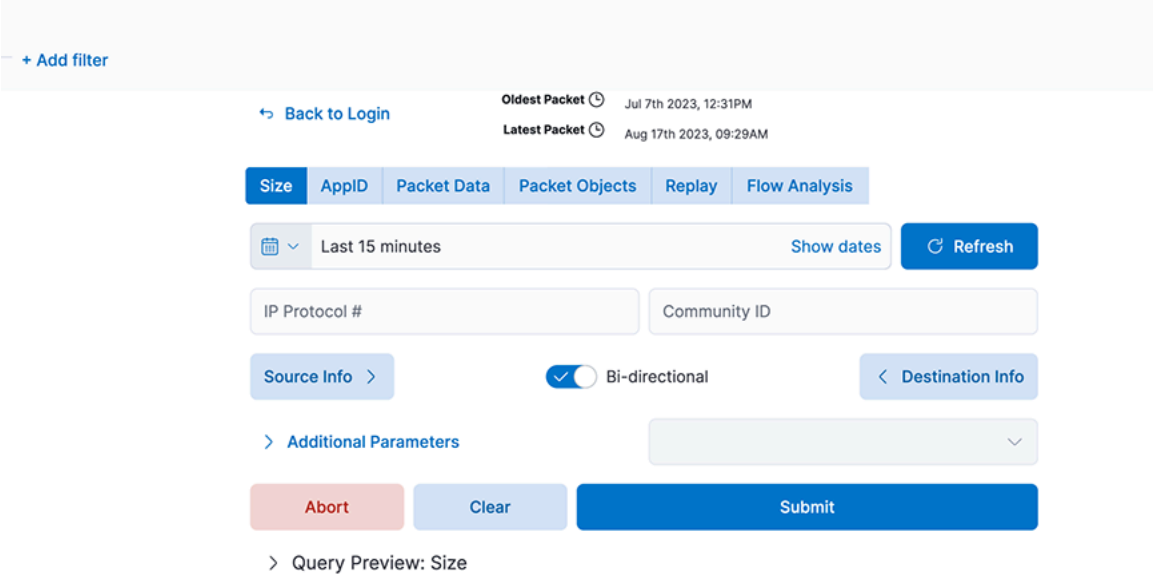
Both NetFlow and TCPflow dashboards have the recorder node visualization.

3.2 General Operation

To retrieve packets from the DMF Recorder Node for analysis using Arista Analytics, select the controller and log in from **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 3-1: DMF Recorder Node



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

You can use the **Recorder Summary** query to determine the number of packets in the recorder database, and then apply filters to retrieve a reasonable number of packets that include the most interesting information.

You can modify the filters in the recorder query until a **Size** query returns the most useful 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 actual 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 search query a URL is generated 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 search query a URL is generated 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 time range for the matching packets either 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 more than one VLAN ID exists.
- **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 returned by the query 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 deselected/none, and the query is sent to all packet recorders configured on the DANZ Monitoring Fabric.
- **Dedup**: Enable/Disable Dedup.
- **Query Preview**: After expanding, this section provides the Stenographer syntax that is 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.

3.3 Using Recorder 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. Use the Summary query to determine the number of packets captured by the Recorder. Apply filters until the packet set is a manageable size (less than ***1*** GB).
2. Search over the metadata received from all sources and analyze 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 DMF Recorder Node 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, Re-transmissions. This analysis is done on various types of packet varieties like HTTP, HTTTP request, DNS, Hosts, IPv4, IPv6, TCP, TCPFlow Health, UDP, RTP Streams, SIP Correlate, and SIP Streams. 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.

   Replay lets you direct large packet sets to an archive for later analysis; this frees up the Recorder to capture a new packet set.

   You can also use DMF Recorder Node for identifying the applications on your network that are encrypting packets. Use a Recorder Detail query to see the applications with encrypted packets.

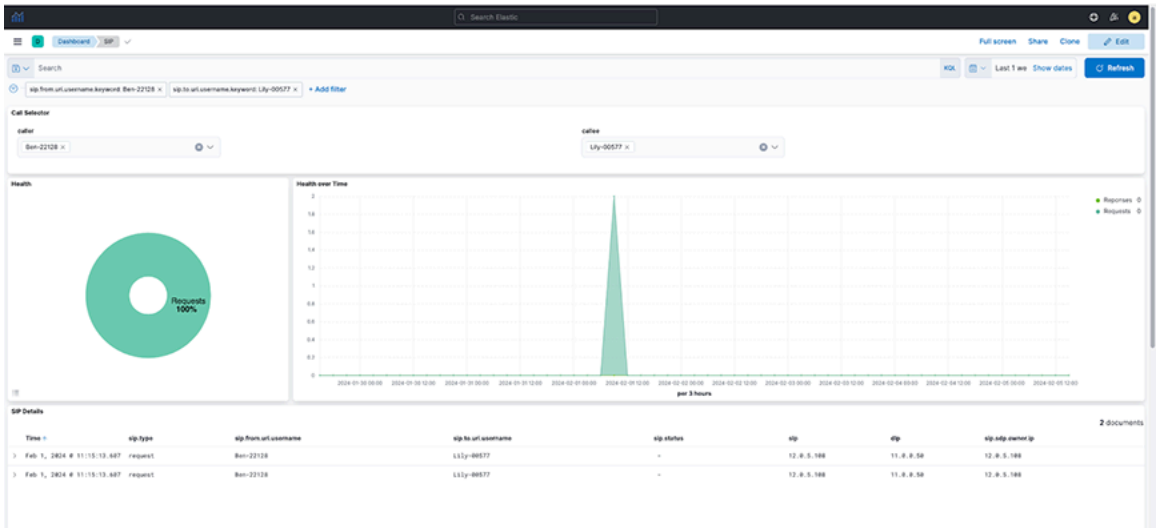
   For information about installing and setting up the DMF Recorder Node, refer to the ***DANZ Monitoring Fabric Deployment Guide***. For details about using the Recorder from the DANZ Monitoring Fabric controller GUI or CLI, refer to the ***DANZ Monitoring Fabric User Guide.***

3.4 Analyzing SIP and RTP for DMF Analytics

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.

The dashboard is located in the dashboard mode of Kibana named **SIP**.

Figure 3-2: SIP Dashboard



DMF Preconditions

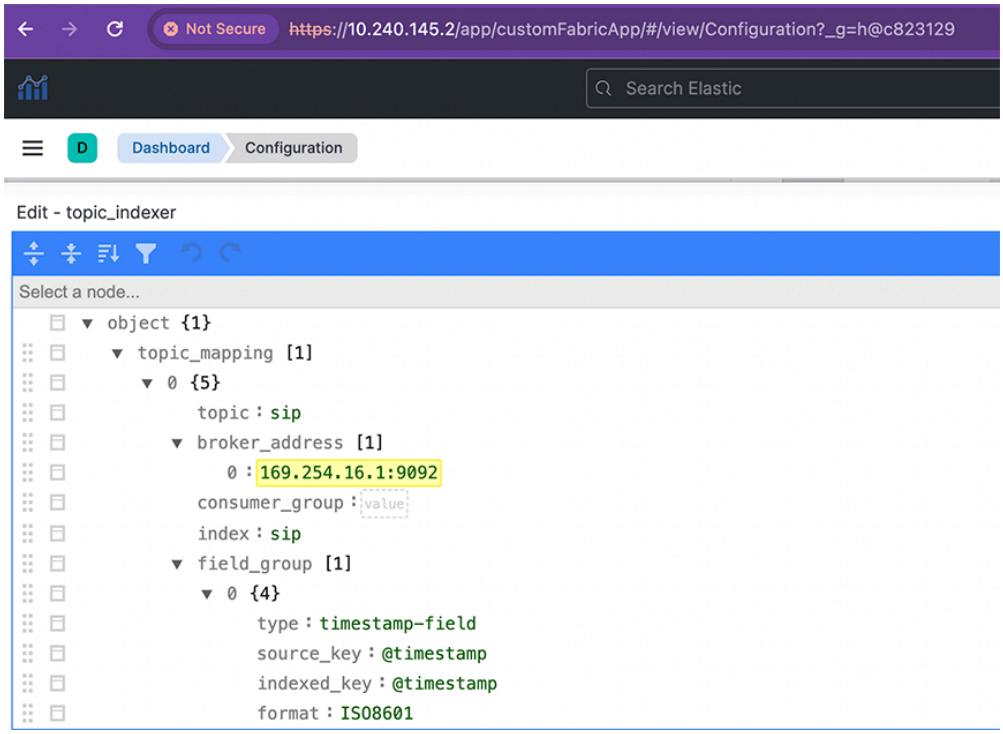
The feature requires a physical connection from the DMF Delivery Switch to the 10G Analytics Node (AN) Collector interface.

- 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 UDF.
- LAG to 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 Field Details for more information on ***broker_address***.

Figure 3-3: Edit-topic indexer



Limitations

This feature is supported in ***AN DMF 8.5.0*** release onwards.

- There is no toggle switch to enable or disable this feature.

Chapter 4

Managing the NetFlow Dashboard

This chapter describes how to manage NetFlow and use the NetFlow dashboard more efficiently. Arista Analytics acts as a NetFlow collector for any agent or generator that is configured with the Analytics server IP address as a collector. This includes DMF Service Node and any third-party NetFlow agent. This chapter has the following sections:

- NetFlow Optimization
- Viewing Filter Interface Information on the NetFlow Dashboard
- Displaying Flows with Out-Discards

4.1 NetFlow Optimization

Arista Analytics may consolidate NetFlow records to improve performance.

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

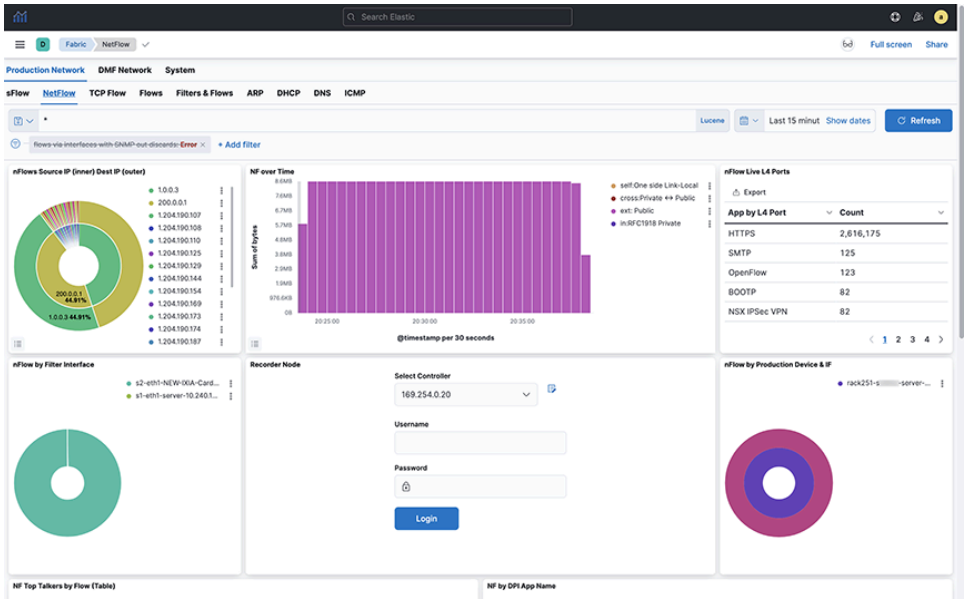
For example, ten flows received by the Analytics server within a thirty second period are consolidated into a single flow if the source and destination IP addresses and destination port are the same for all the flows and only the source ports are different, or if the source and destination IP addresses 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 enable the Netflow Optimization for Netflow v9 and IPFIX please refer to Consolidating Netflow V9/IPFIX records section.

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

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

Figure 4-1: Analytics NetFlow Optimization



4.2 Viewing Filter Interface Information on the NetFlow Dashboard

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. It accomplishes by querying a specific flow. Arista Analytics then shows the filter interface name associated with that flow. If the flow goes through two hops, then two filter interface names are displayed for the flow.

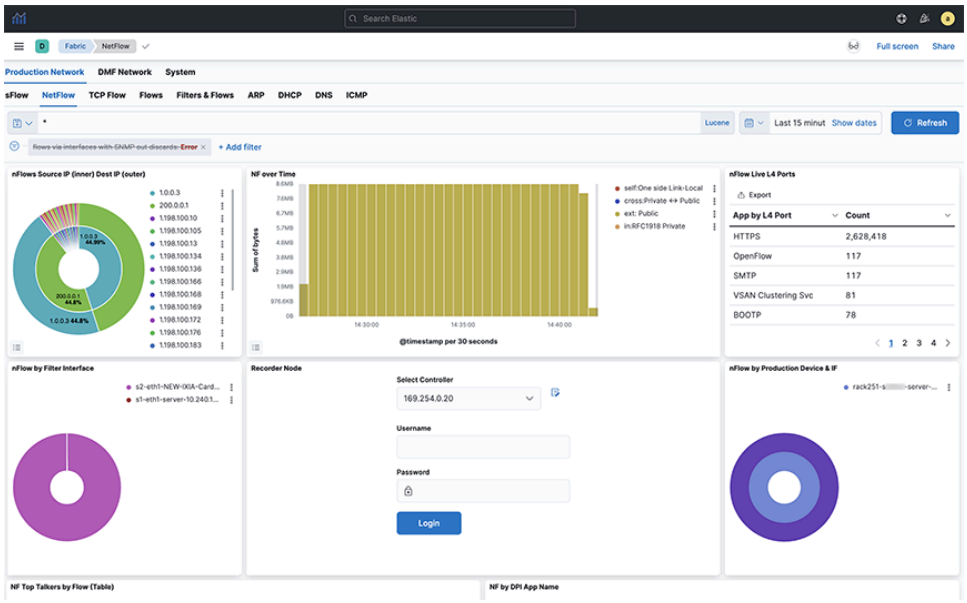
4.2.1 Displaying Filter Interface Names

The **nFlow by Filter Interface** window on the **NetFlow** dashboard, shown below, 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-interface
```

Figure 4-2: Production Network > NetFlow Dashboard with Filter Interface Name



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

NetFlow Managed Service Records-per-interface Option

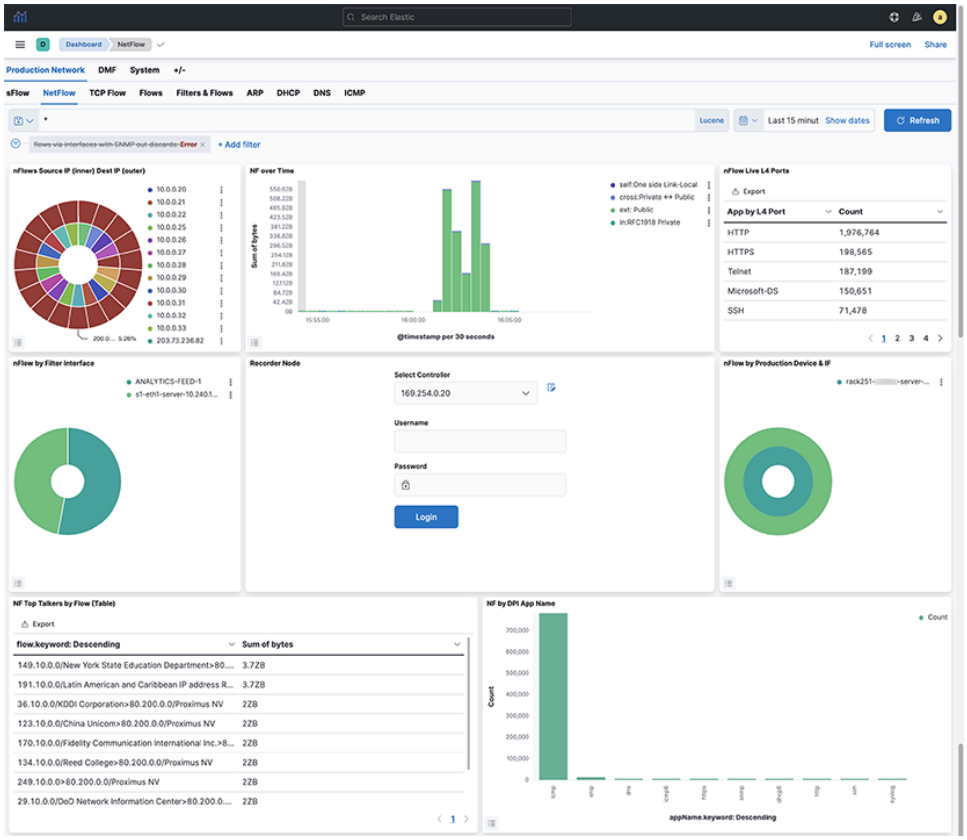
```
! managed-service
managed-service netflow-managed-service
 service-interface switch 00:00:4c:76:25:f5:4b:80 ethernet4/3:4
 !
 service-action netflow netflow-delivery-int
  collector 10.8.39.101 udp-port 2055 mtu 1500 records-per-interface
controller(config)# sh running-config bigtap policy netflow-policy
! 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 above, 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 connected to the filter interface in the **nFlow by Production Switch & IF** window.

In the example below, ***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, shown below.

Figure 4-3: Production Network > NetFlow Dashboard with Filter Interface Name



4.2.2 NetFlow Traffic Coming from Third-party Devices

This section displays third-party device and interface names. It lets you see hop-by-hop forwarding of flows when NetFlow traffic is coming from a third-party device. When you query 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 on which NetFlow is enabled on each 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: From ***8.3.0***, both SNMPv2 and SNMPv3 are supported.

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, the

iFindex is not sent 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.

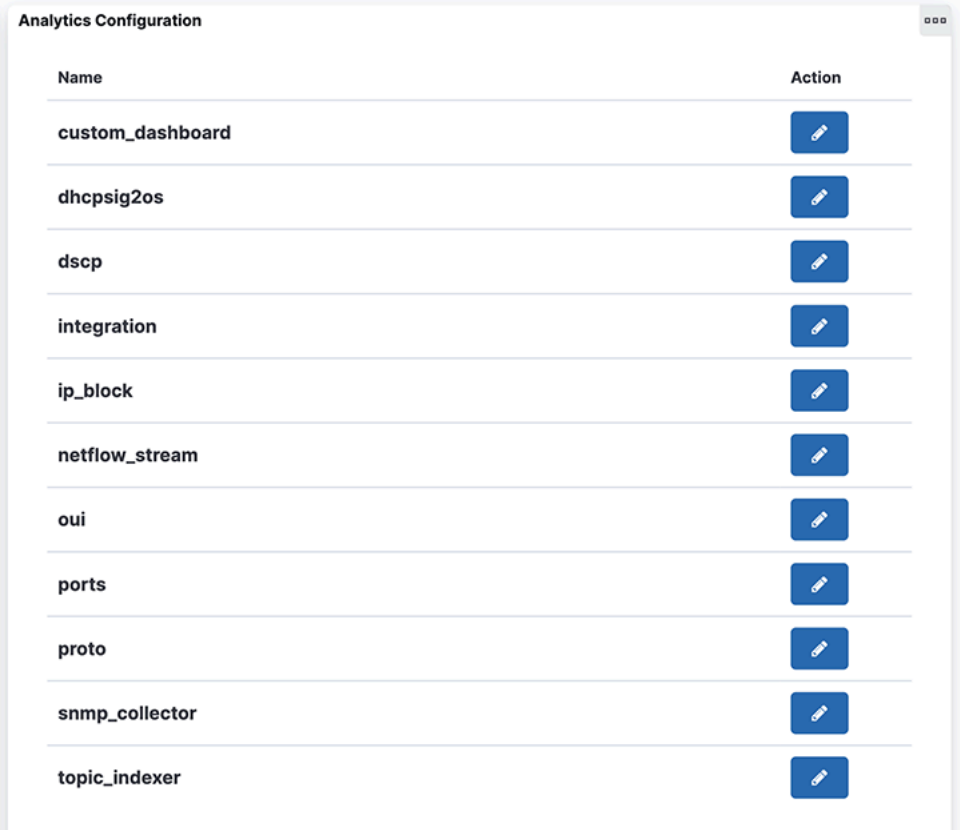
DMF Analytic > **System** > **Configuration** > **Analytic Configuration** > **snmp_collector**

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 below, click **DMF Analytic** > **System** > **Configuration** > **Analytic Configuration** > **snmp_collector** > **Edit**.

   Figure 4-4: Analytic snmp_collector config



The system displays the following edit dialog.

Figure 4-5: Analytic Configuration > snmp_collector > Edit Dialog (SNMPv2 Configuration)

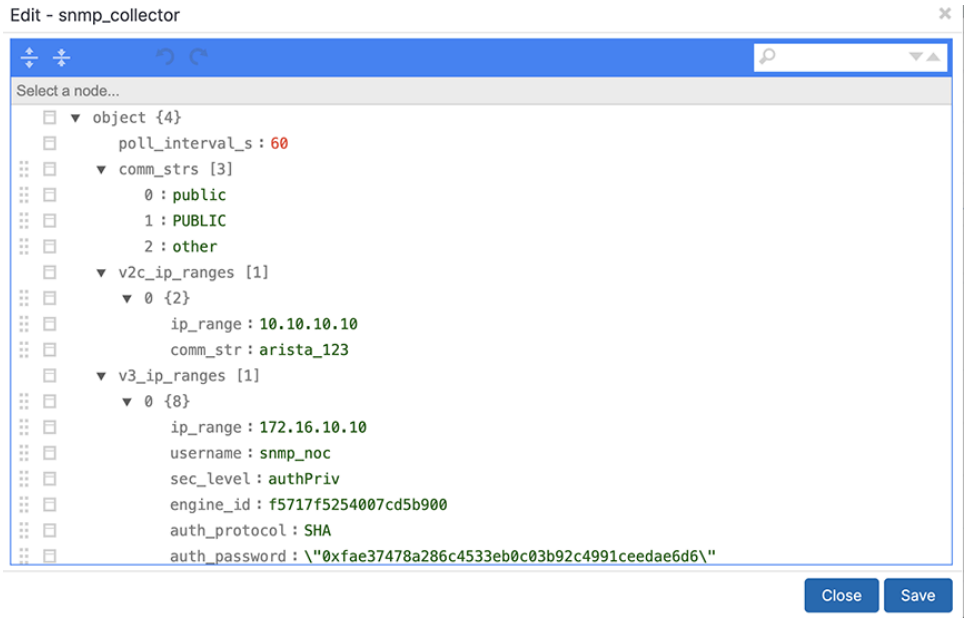
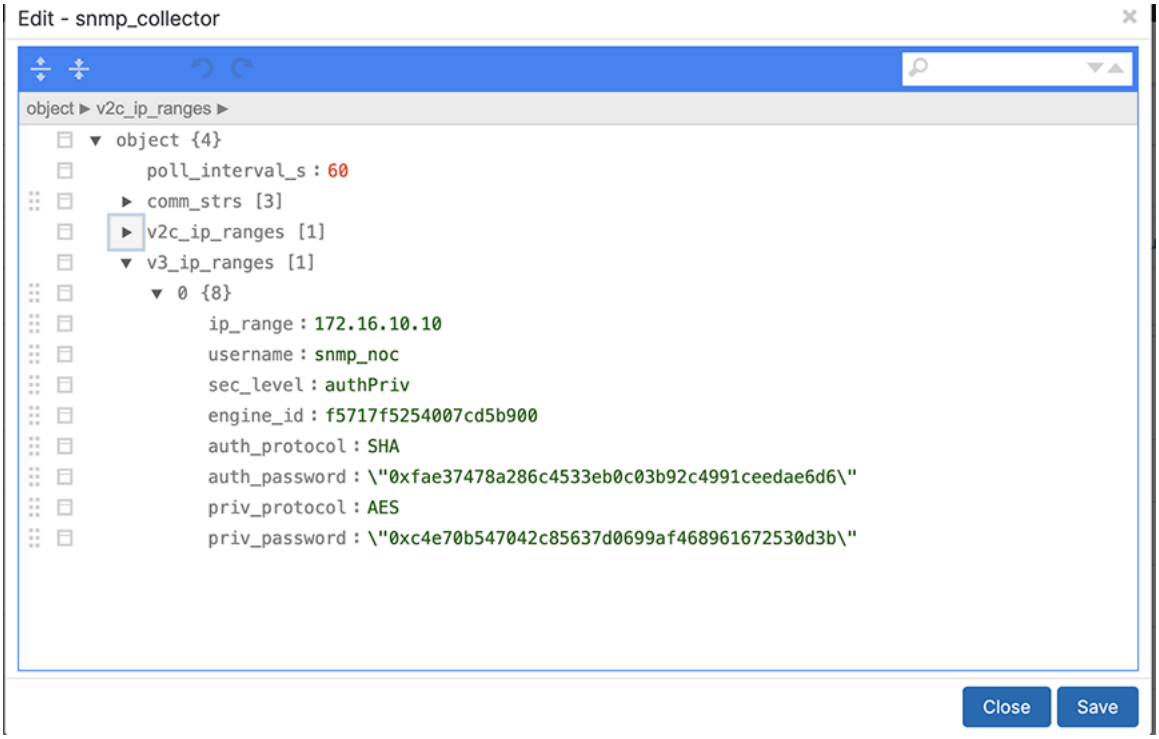


Figure 4-6: Analytic Configuration > snmp_collector > Edit Dialog (SNMPv3 Configuration)

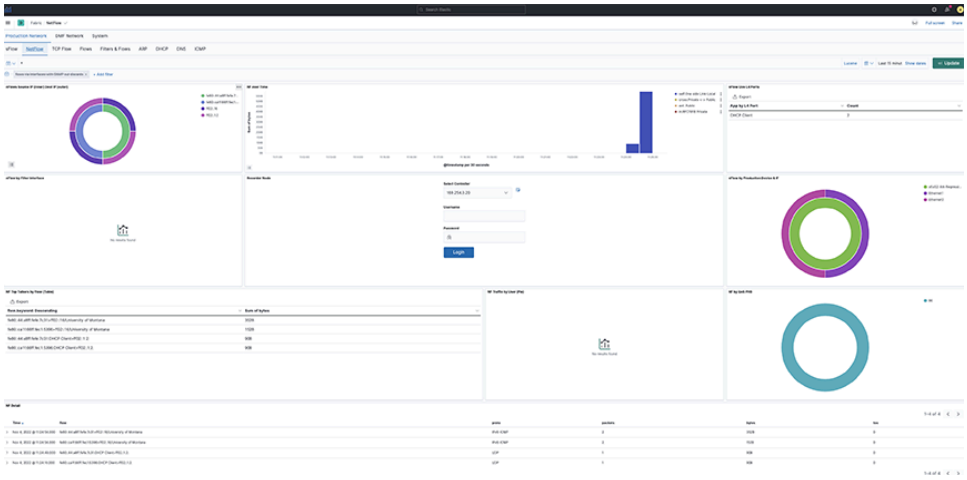


2. Click the community string **public** to change it to a different value as shown in the following dialog.

   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 and it is displayed in the **nflow by Production Device & IF** window.

Figure 4-7: Analytic Configuration > snmp_collector > Edit Dialog



4.3 Displaying Flows with Out-Discards

The **NetFlow** dashboard provides an option to display flows with out-discards when the NetFlow packets come from third-party devices. To display this information, use 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.

Chapter 5

Monitoring DMF Network Health

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

5.1 DMF Network Tab

The **DMF Network** tab includes dashboards that display the following information visible to the DMF controller:

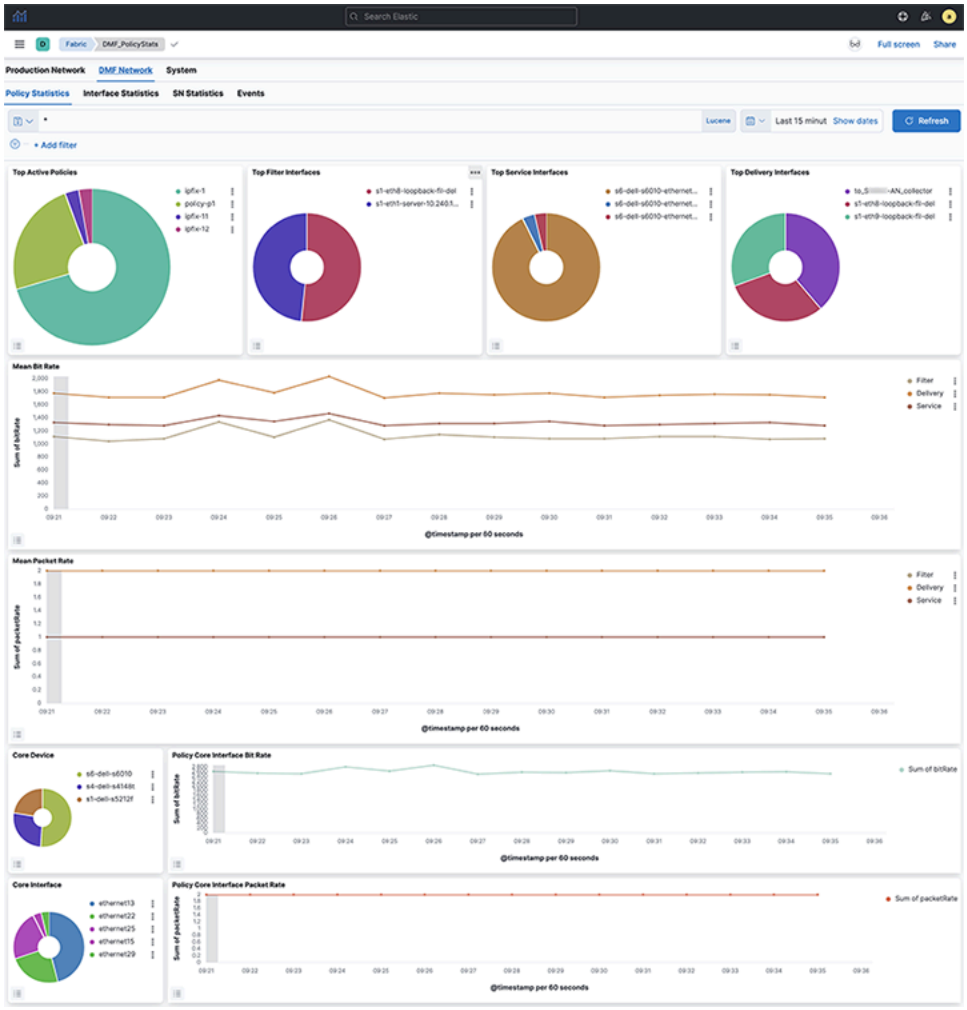
- Policy Statistics
- Interface Statistics
- SN Statistics
- Incline Statistics
- Events

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

5.2 Policy Statistics Dashboard

When you click the **Policy Statistics** tab, the system displays the following dashboard:

Figure 5-1: DMF Network > Policy Statistics Dashboard



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

- Top Active Policies
- Top Filter Interfaces
- Top Service Interfaces
- Top Delivery Interfaces
- Mean Bit Rate
- Mean Packet Rate
- Core Switch
- Policy Core Interface Bit Rate
- Core Interface
- Policy Core Interface Packet Rate
- Records
- Policies with no traffic

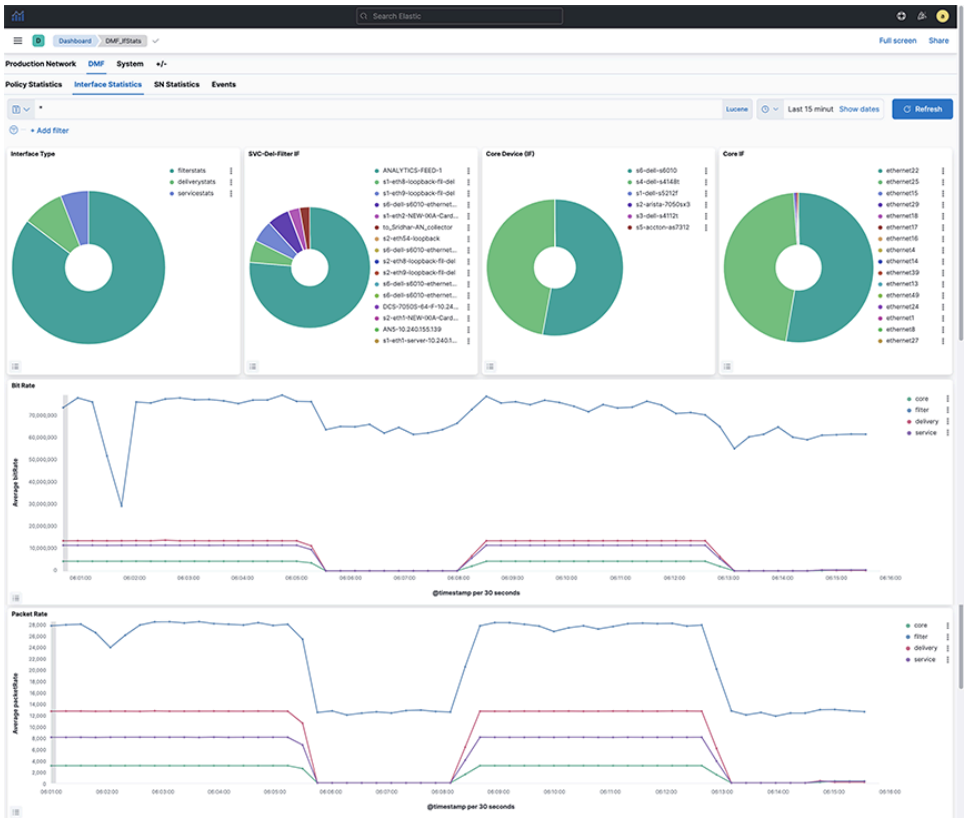
You can use the **Top Active Policies** visualization to verify that your DANZ Monitoring Fabric policies are active and behaving as expected.

You can use the **Filter Interfaces** visualization to balance the utilization of your filter interfaces and to help make sure no packets you want to analyze are dropped.

5.3 Interface Statistics

When you click the **Interface Statistics** tab, the system displays the following dashboard.

Figure 5-2: DMF Network > Interface Statistics



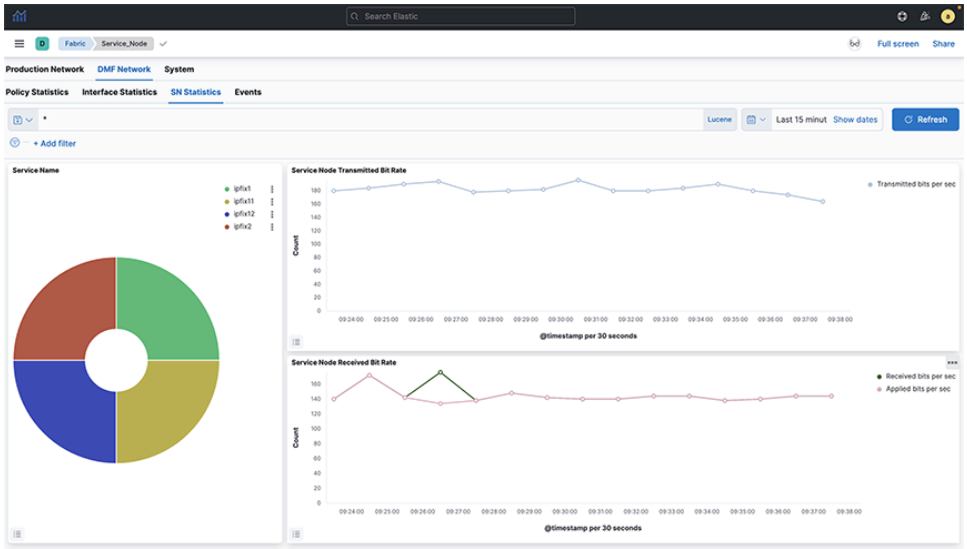
The **Interface Statistics** dashboard summarizes information about DANZ Monitoring Fabric switch interface activity and provides the following panels:

- Interface Type
- SVC-Del-Filter IF
- Core Switch (IF)
- Core IF
- Bit Rate
- Packet Rate
- Interface Detail

5.4 SN (Service Node) Statistics

When you click the **SN Statistics** tab, the system displays the following dashboard.

Figure 5-3: DMF Network > SN Statistics

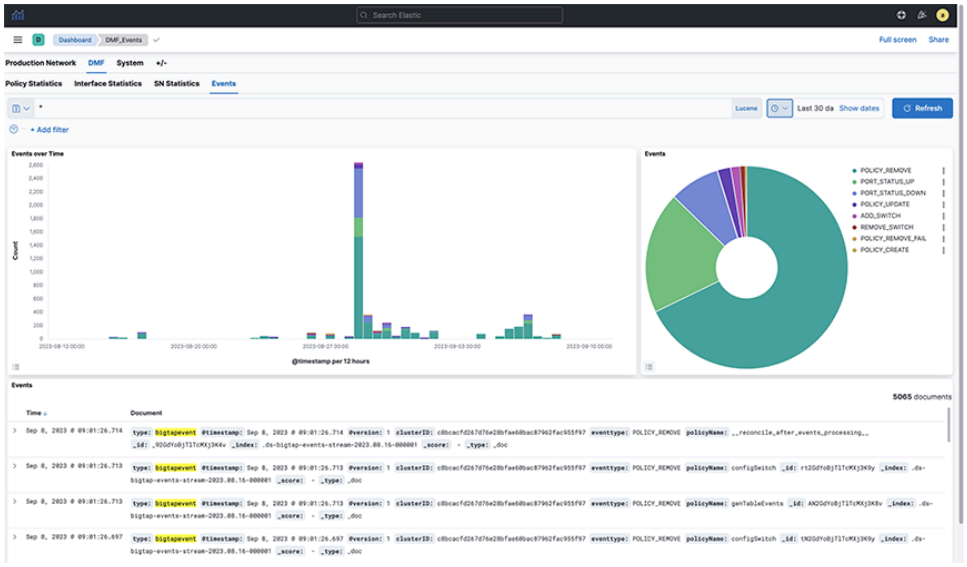


To see the statistics for a specific managed service, select a service from the pie chart. This will display statistics for the selected service.

5.5 Events

When you click the **Events** tab, the system displays the following dashboard.

Figure 5-4: DMF Network > Events



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

- Events Over Time

- Events

Chapter 6

Monitoring Users and Software Running on the Network

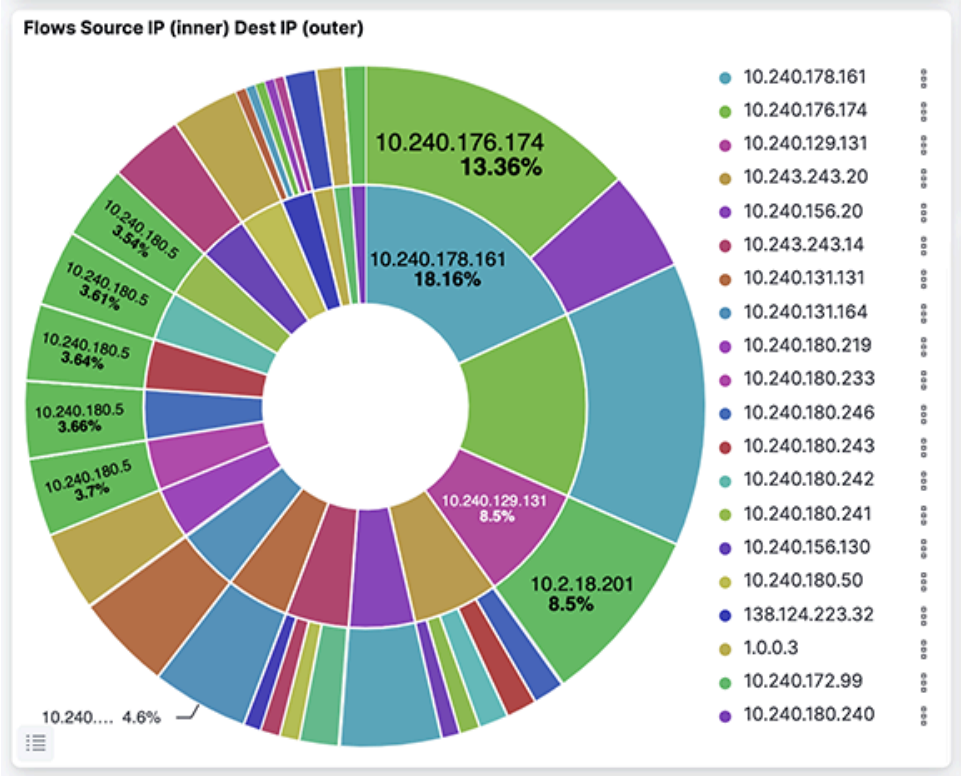
This chapter describes how to use Arista Analytics with the DMF Recorder Node. It includes the following sections.

6.1 IP Addresses

This section describes how to identify traffic transmitted or received by the source or destination IP address.

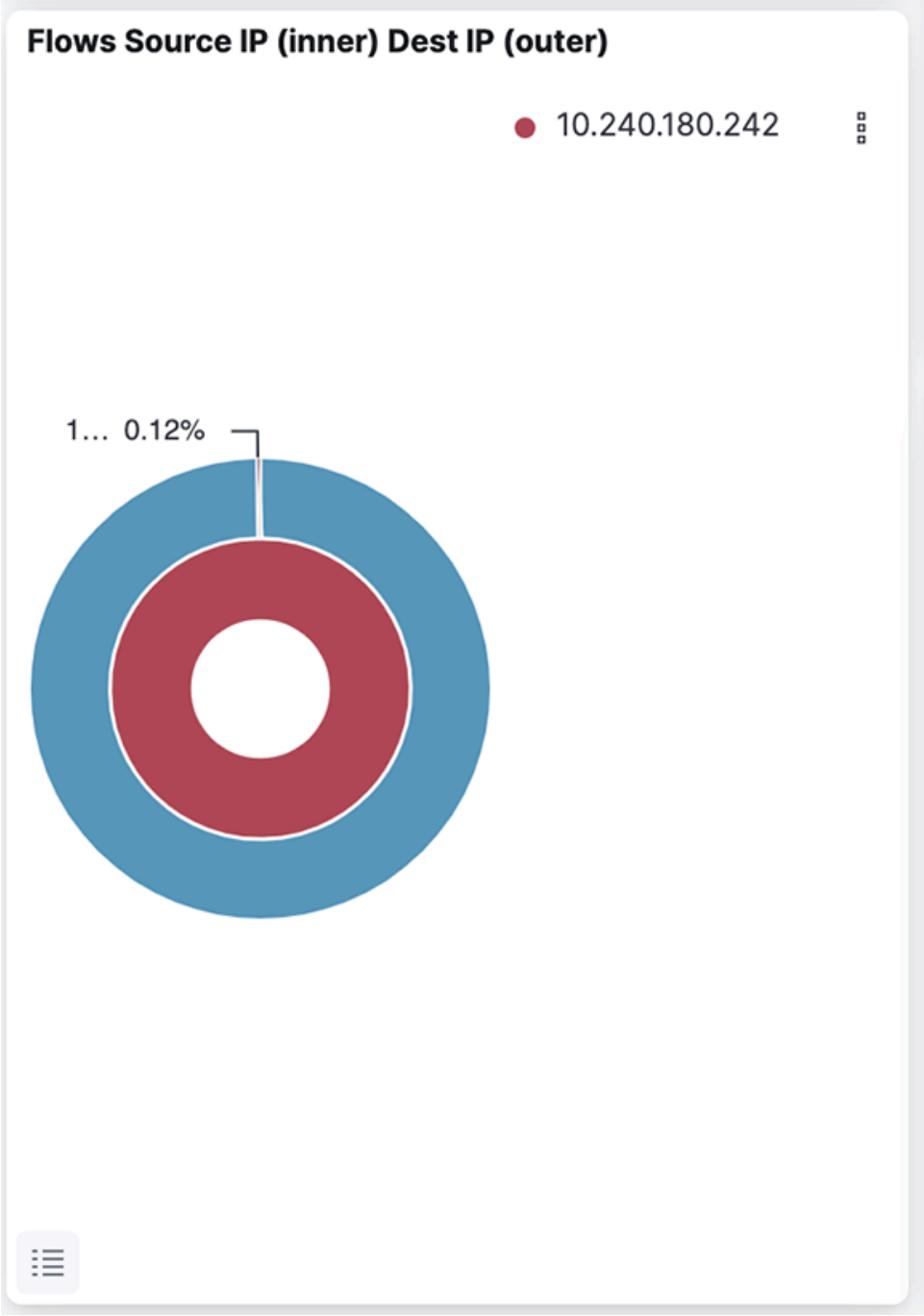
6.1.1 Source and Destination Addresses

Figure 6-1: Identifying Source and Destination IP Addresses



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

Figure 6-2: Filtering Results by IP Address



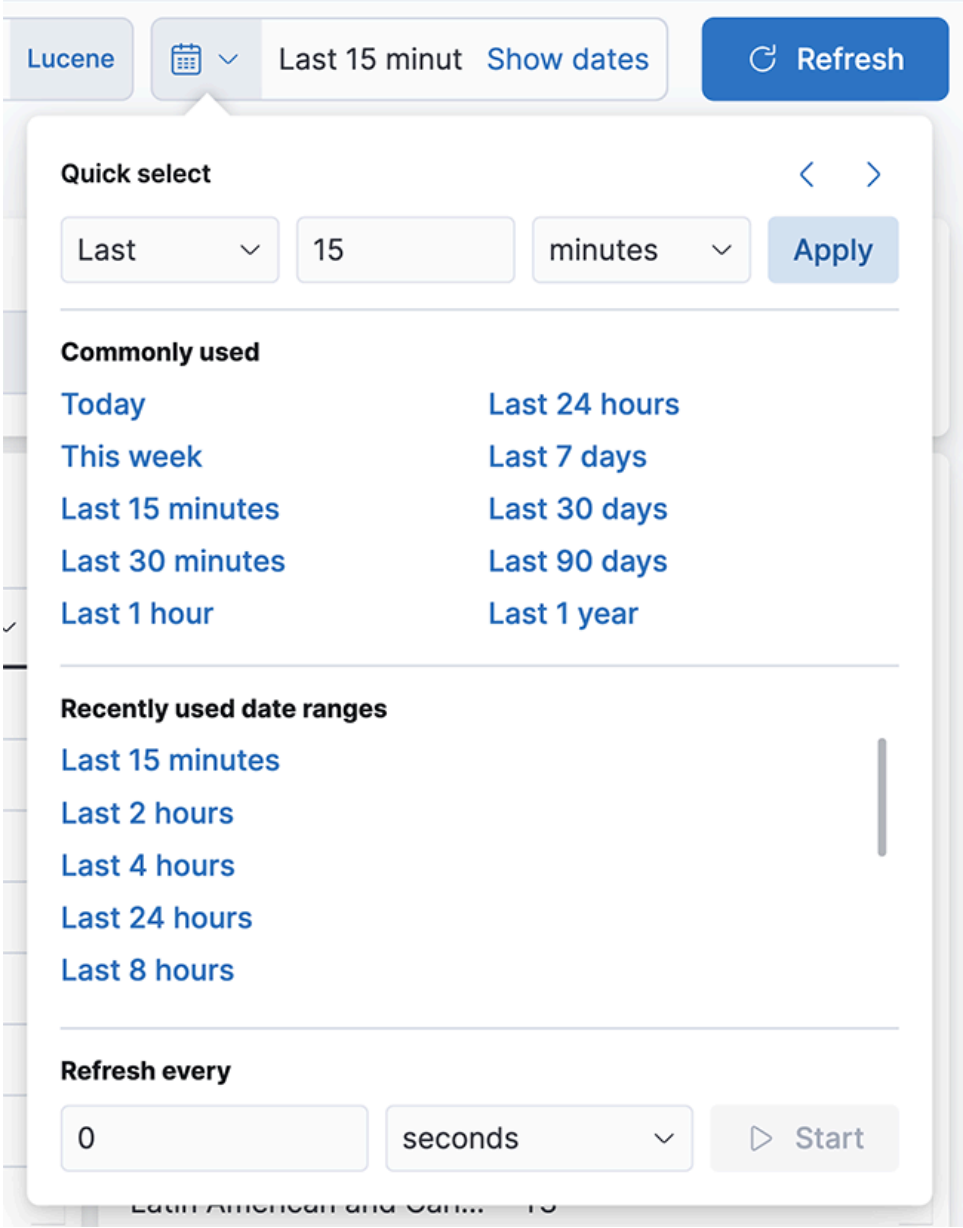
The selected IP address is added to the filters on the dashboard.

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

6.1.2 Unauthorized IP Destinations

To determine if an IP destination that is not authorized is being accessed in your network for a specific period of time, set the time value in the upper right corner.

Figure 6-3: Setting the Duration



Select the duration you want to search.

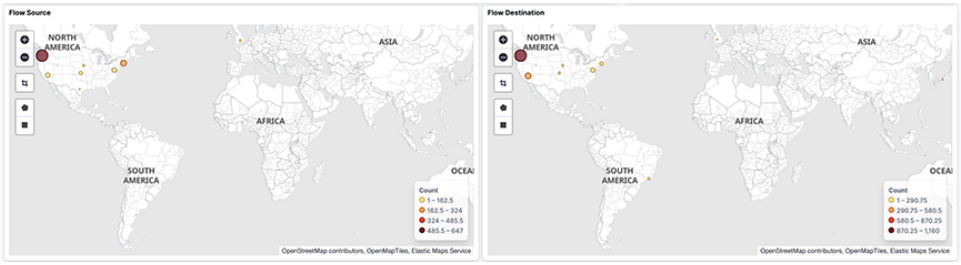
Type the IP address or the Network ID in the **Search** field.

The system displays any events associated with the address or network ID.

6.2 Geographic Location

Analytics associates public network IP addresses to geographic regions using the MaxMind GeoIP database. Traffic associated with these addresses is shown as a heat map on the **Map** visualization on the **sFlow** dashboard. You can filter on a region by drawing a box or a polygon around the region.

Figure 6-4: Geographic Flow Source and Destination



Use the **Square** tool to draw a square around a region of interest, or use the **Polygon** tool to draw an irregular shape around a region. The map is redrawn to zoom in on the selected region and to show details about traffic to or from the region.

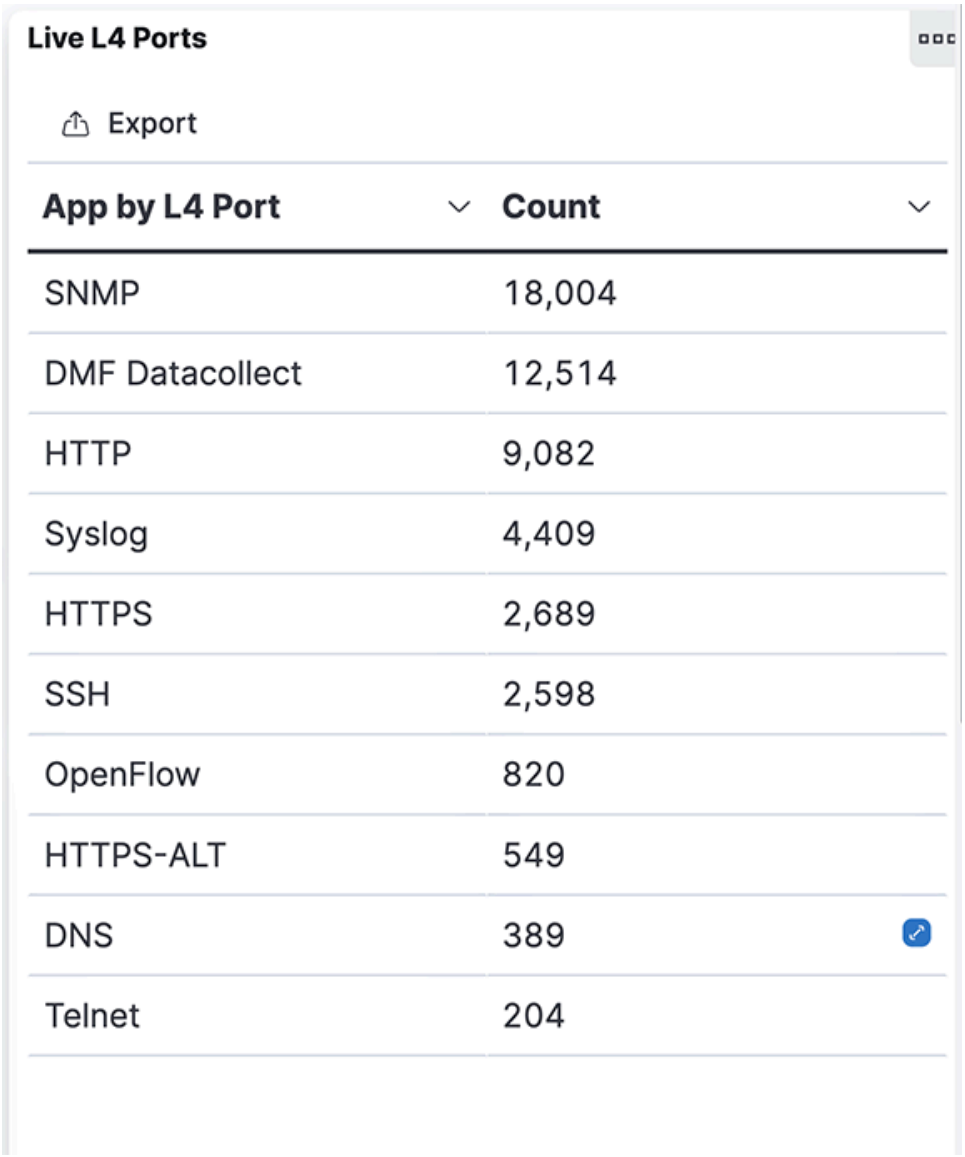
6.3 Software Running in the Network

This section describes how to identify specific applications or operating systems running on network hosts.

6.3.1 Top Talkers Using Well-known Layer-4 Ports

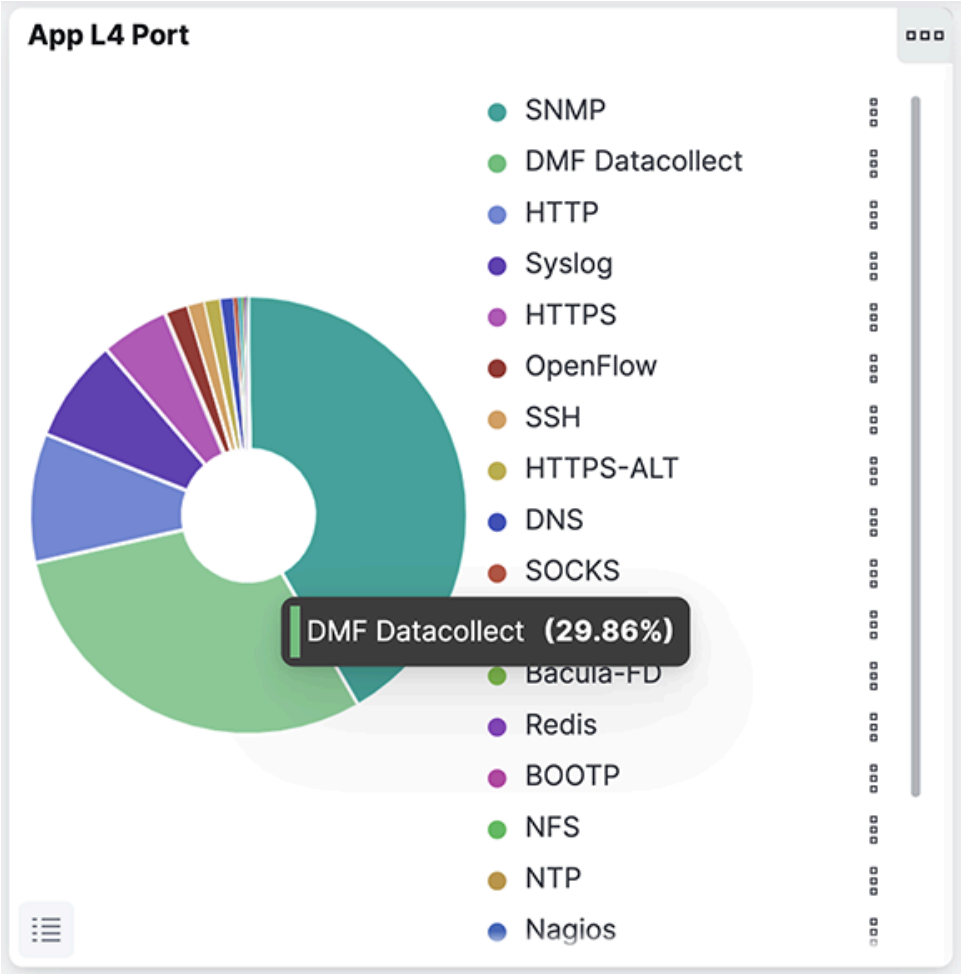
To view top-N statistics for the flows using a well-known L4 port, use the **Live L4 Ports** table on the **Flows** dashboard.

Figure 6-5: Flows > Live L4 Ports



If you have an sFlow generator configured to send flows to Analytics, you can also use the **App L4 Port** table on the **sFlows** dashboard.

Figure 6-6: sFlow > App L4 Port



These tables use well-known ports to identify the traffic generated by each application. You can also associate user-defined ports with applications as described in the following section.

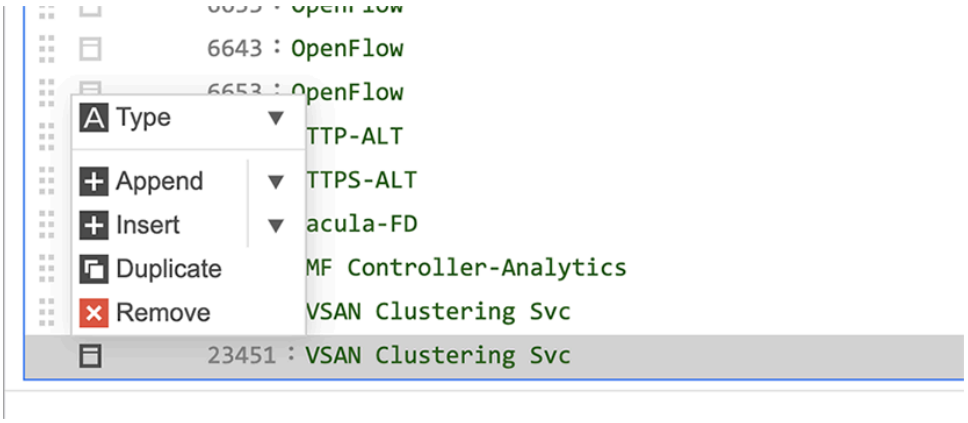
6.3.2 Associating Applications with User-defined Layer-4 Ports

To associate user-defined ports with applications, complete the following steps:

1. Select **System** > **Configuration**.

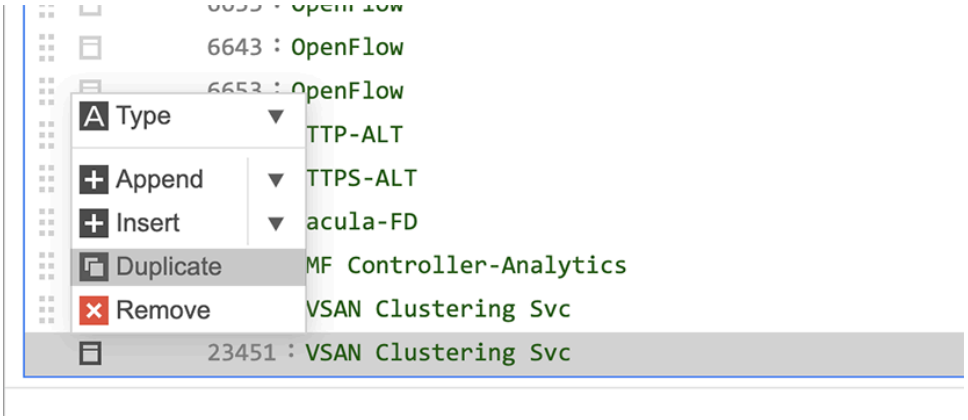
2. Select the **Edit** control to the right of the **Ports** section.

Figure 6-7: Edit Ports



3. To copy an existing row, enable the checkbox to the left of the row and select **Duplicate** from the drop-down menu.

Figure 6-8: Duplicate Ports



4. Type over the port number in the row you copied and enter an associated label.

   For example, you could assign ***port 1212*** to ***Customer App X***.

5. Click **save**.

6.3.3 Software Running on Hosts

To identify the software running on hosts in the monitored network, you can use the following features.

- Searching for well-known applications
- Using Layer-4 labels
- Searching packet captures on the DMF Recorder Node
- Using the **Flows** dashboard
- Using the **DHCP** dashboard for information about operating systems

The IP block default mapping associates many common applications with specific address ranges. For example, you can identify video traffic by searching for ***YouTube*** or ***Net#ix***.

L4 label strings identify applications using well-known ports, and applications running on user-defined ports after you map those ports to the applications.

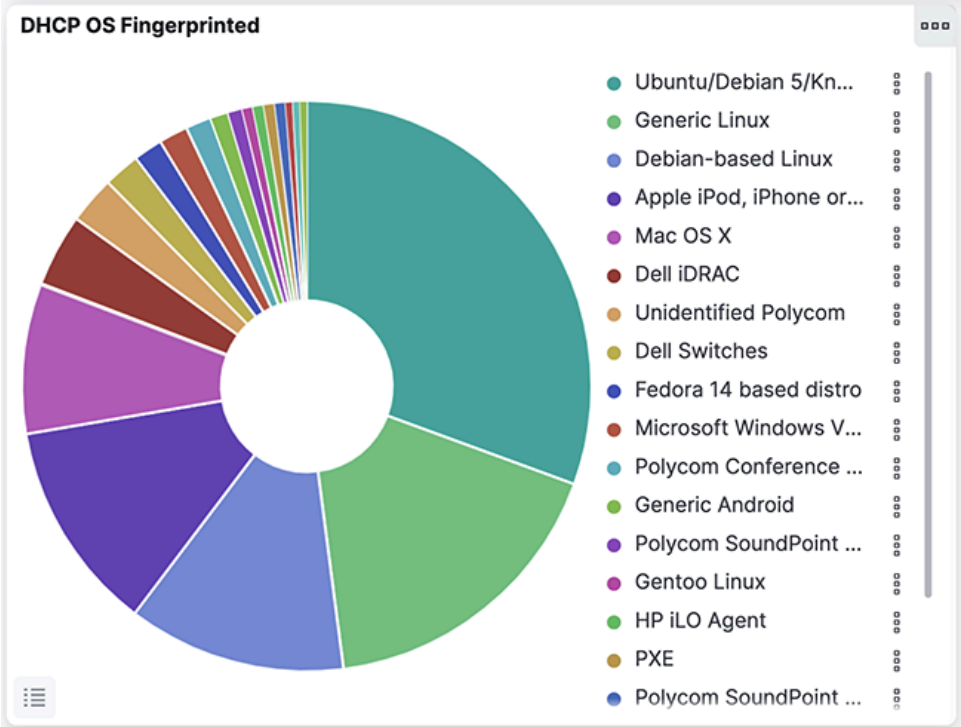
The flow dashboards all give an overall sense of who is talking to whom. Click on an IP address or L4 port and with the **+** that appears and pin that to filter the dashboard by the selection. Every dashboard has a bar

chart depicting traffic in the y-axis and time in the x-axis. Note that a time filter can be added by a click and sideways select of the bar chart.

The who can also be in terms of the user with a source of users to IP mappings (OpenVPN supported) configured. After that, a search by the user string can be carried out to see traffic attributed to that user over the period of the dashboard.

The DHCP dashboard indicates the operating systems running on hosts based on information derived from DHCP client requests. The default mapping is copied from the signatures provided by **`fingerbank.org`**.

Figure 6-9: DHCP OS Fingerprinting



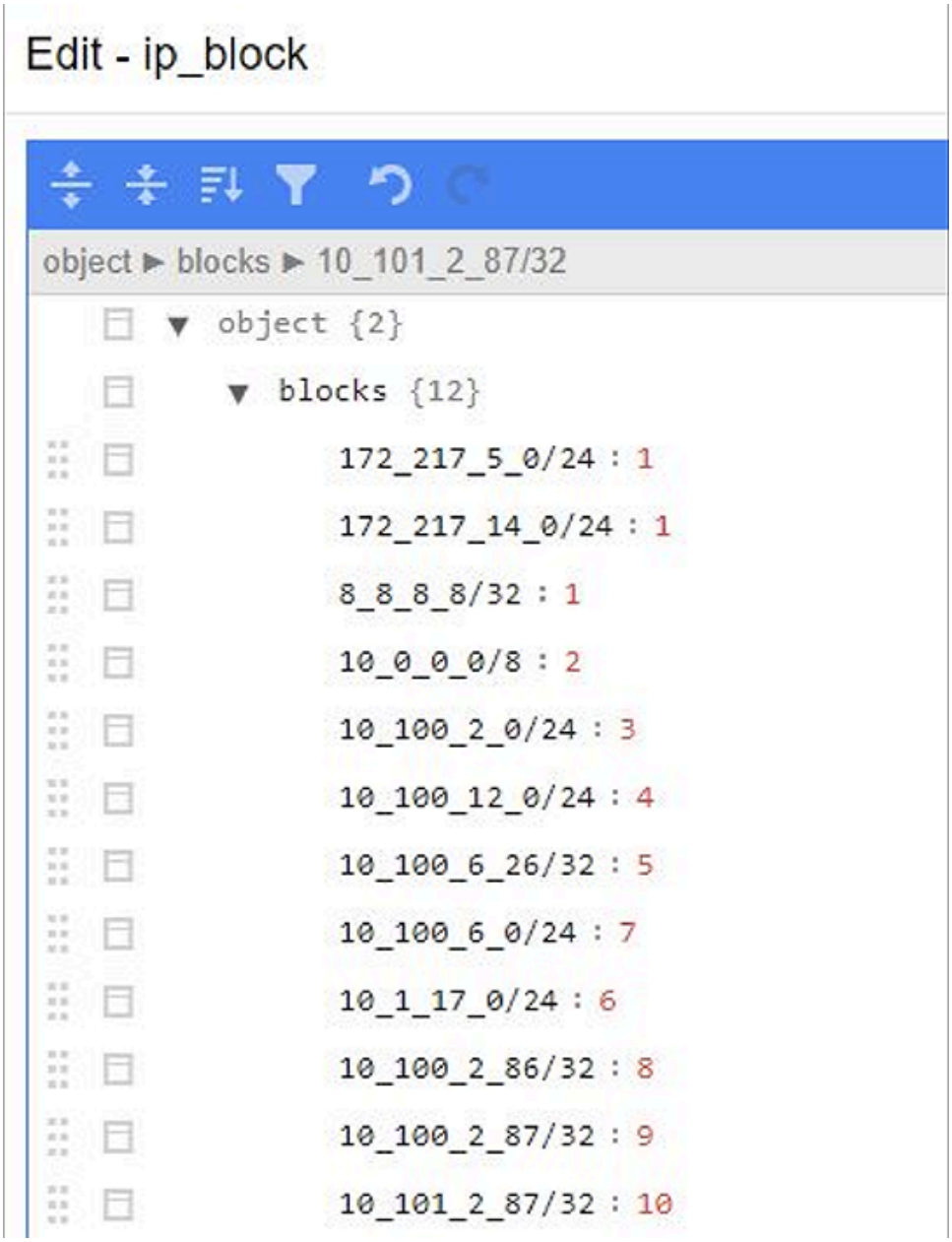
6.3.4 Tools Receiving Traffic

You can identify traffic forwarded to a specific tool or host by using the IP Blocks mapping to associate an IP address or a range of IP addresses to a label describing the application. This label will then appear on any dashboards or visualizations that display the IP Block labels. After mapping, you can also search for events associated with the label assigned to the tool.

For details about updating the IP blocks mapping file, refer to the Mapping IP Address Blocks section.

1. To edit the IP blocks, select **System** > **Configuration** and click the **Edit** control to the right of the IP blocks section.

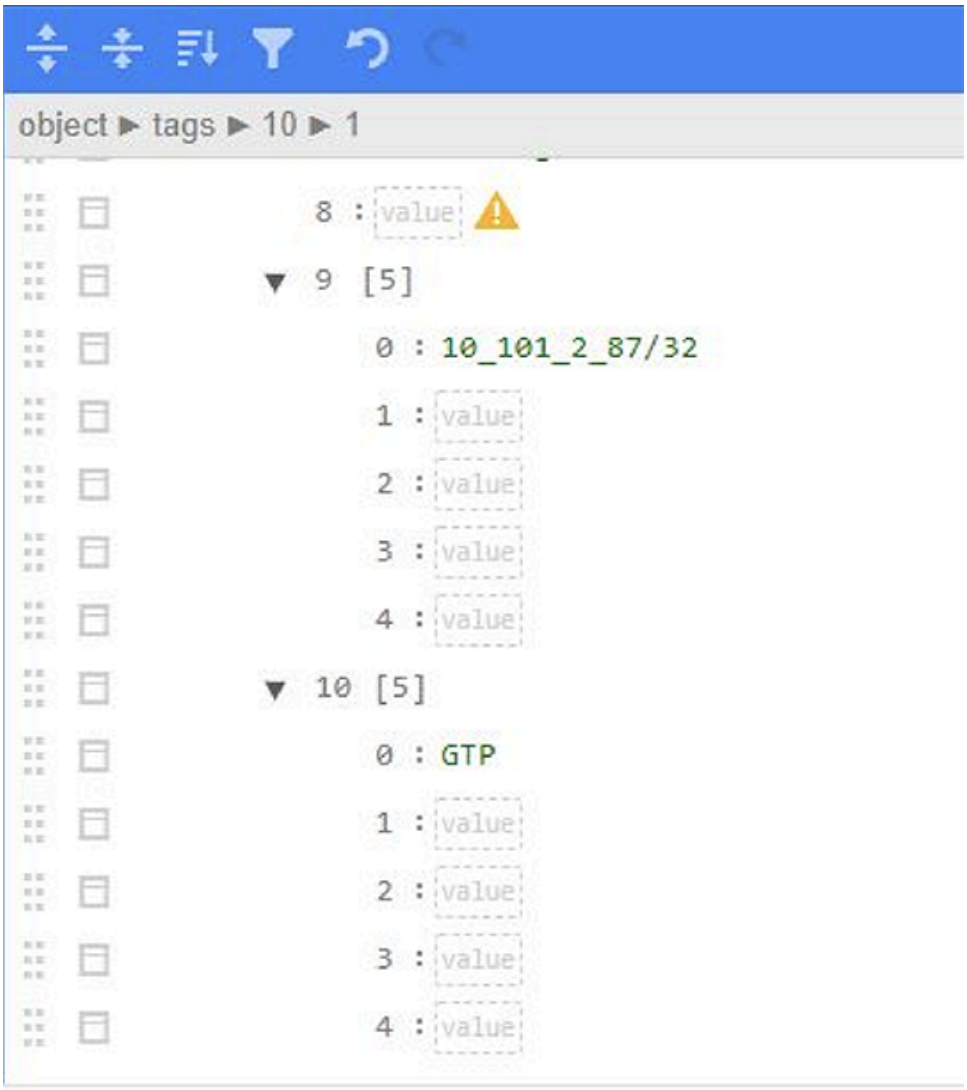
   Figure 6-10: Mapping a Tool to an IP Address: IP Block Edit



2. To define a new IP block, append a range of IP addresses to the blocks section.

3. Scroll down and add a tag definition with the same number as the IP block.

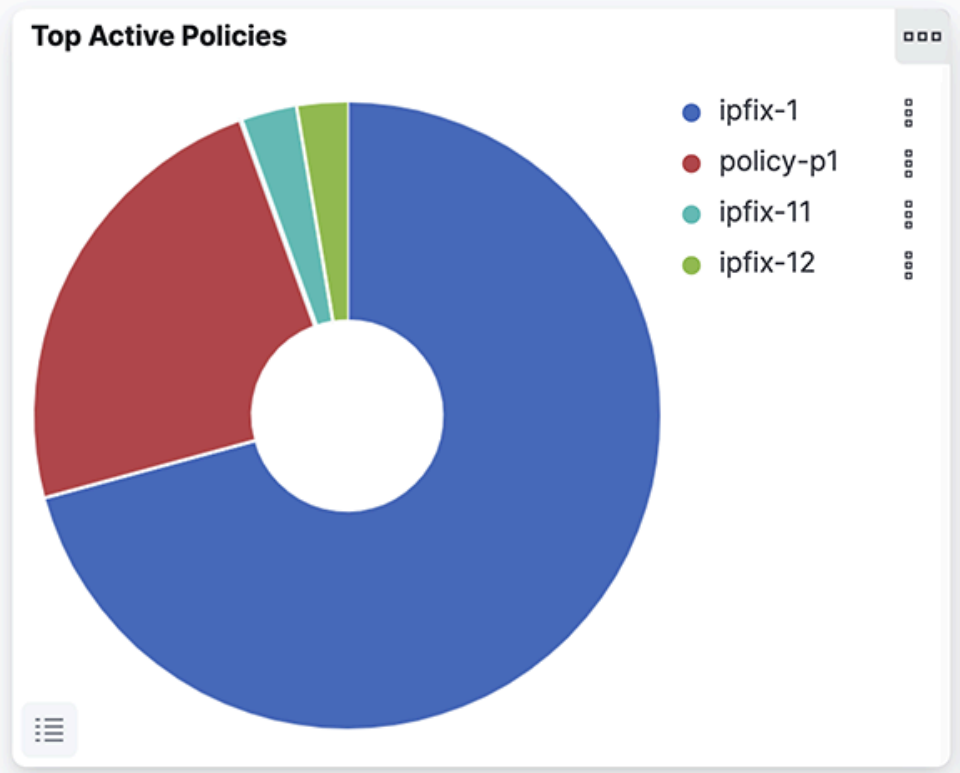
Figure 6-11: Mapping a Tool to an IP Address: Define Tags



4. Define the tags for the new IP block section, including a descriptive name for the specific tool.
5. Select **DMF Network** > **Policy Statistics**.

You can cross reference the information you get by labeling an IP block with information about any policies that are configured to forward traffic to that IP address.

Figure 6-12: DMF Policies



6.4 User Activity

This section describes how to identify specific users transmitting or receiving traffic on the network.

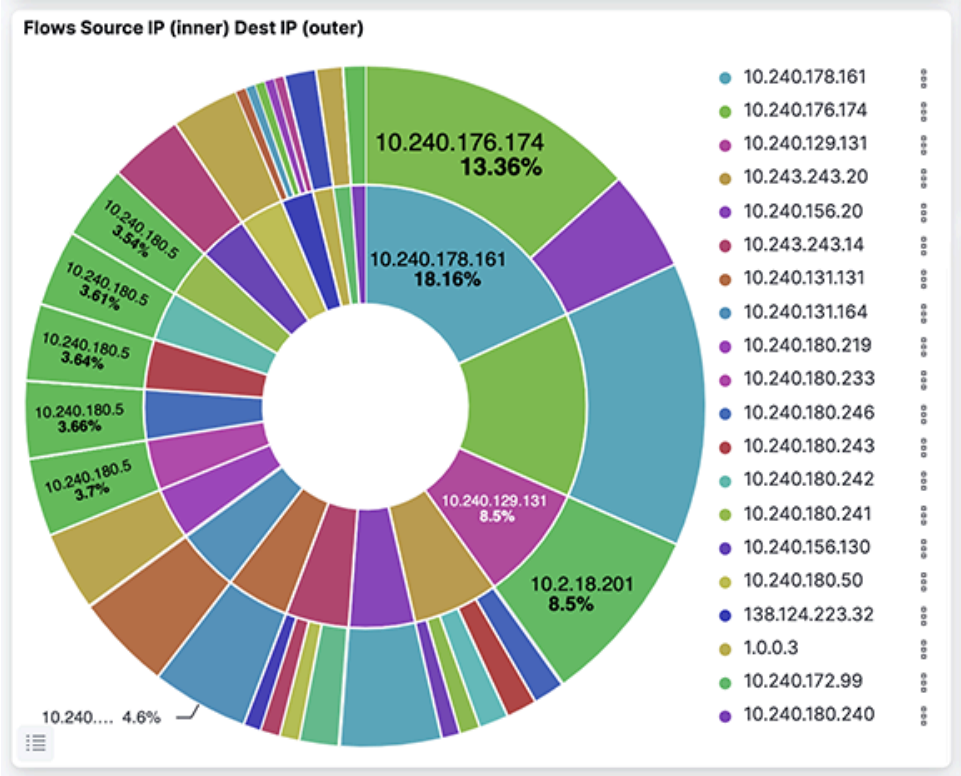
6.4.1 User Sessions

To identify users transmitting or receiving traffic on the network, use the following features:

- **Flows** dashboard
- **sFlow** dashboard
- **NetFlow** dashboard
- Open VPN or Active Directory mapping to IP address

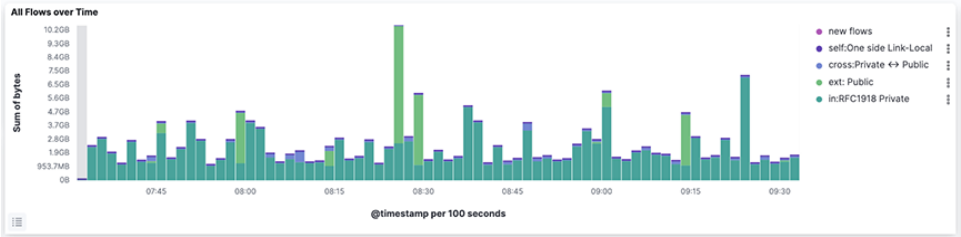
The **Flows** dashboards all provide an overall idea of who is communicating on the network (traffic source and destination).

Figure 6-13: Flows > Flows Source IP Dest IP



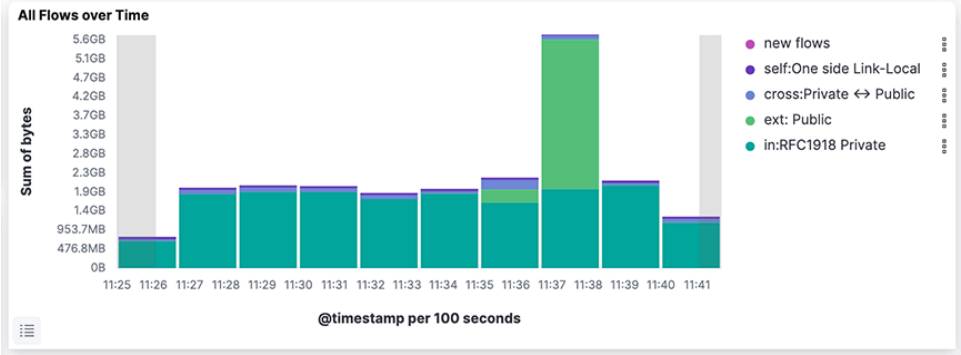
Click an IP address or L4 port and with the **+** that appears, pin that to filter the dashboard for the selection. Every dashboard has a bar chart that shows traffic in the y-axis and time in the x-axis.

Figure 6-14: All Flows Over Time



To filter the display to a specific time period, click and drag from left to right over the interesting time period.

Figure 6-15: All flows Over Time (Specific Time)



You can also identify traffic associated with specific users after using the IP blocks configuration to map the users to a specific IP address. Once it is saved you can search for the user string to see traffic attributed to that user over the period of time displayed on the dashboard.

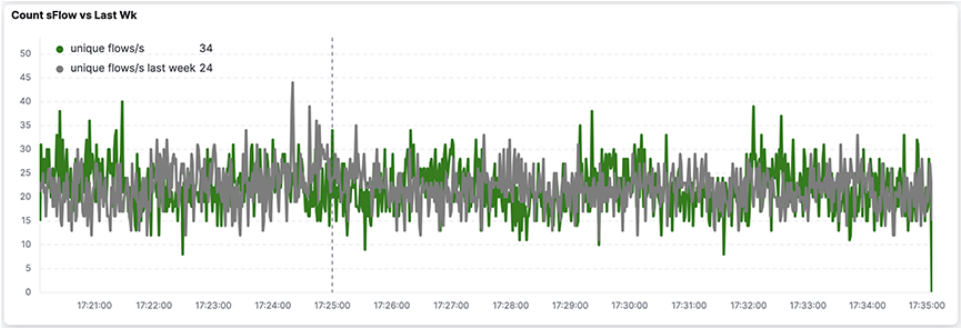
6.4.2 New Network Users

To identify new network users, use the following features:

- Comparing the same dashboard for two different time periods
- **sFlow > Count sFlow vs Last Wk**
- **ARP** dashboard
- New Host Report

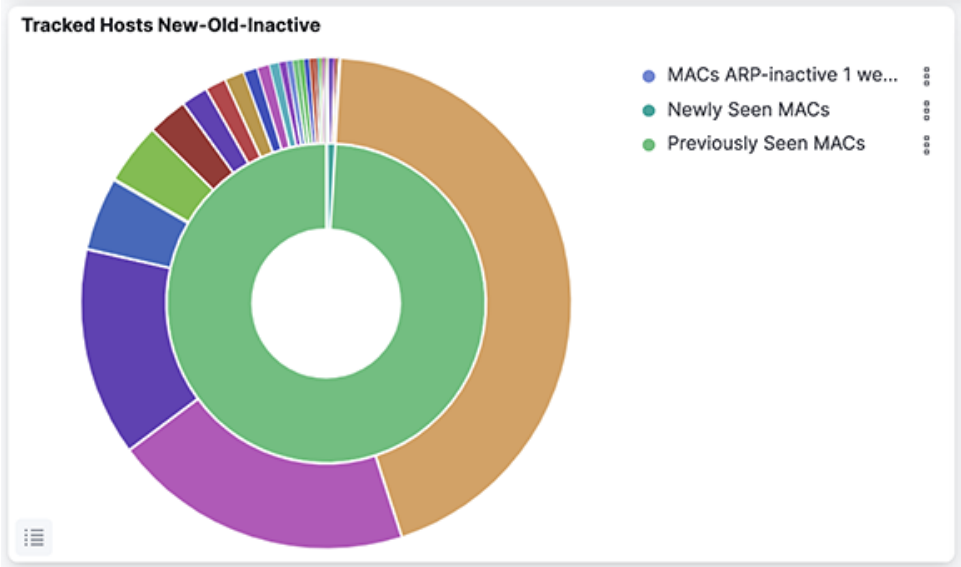
The sFlow dashboard provides a **Count sFlow vs Last Wk** visualization, which shows the number of unique flows being seen now vs. last week

Figure 6-16: sFlow > Count sFlow vs Last Wk



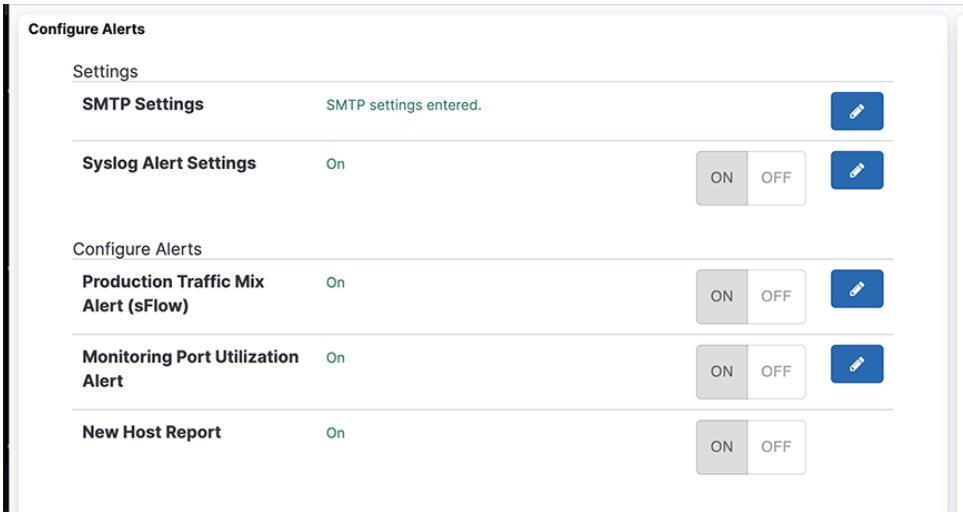
The **ARP** dashboard provides a visualization for **Tracked Hosts New-Old-Inactive, Vendor**.

Figure 6-17: ARP > Tracked Hosts New-Old-Inactive, Vendor



To use the **New Host** report, enable the report and configure where to send alerts on the **System** > Configuration page.

Figure 6-18: System > Configuration > New Host Report



6.4.3 Unauthorized Intranet Activity

To identify unauthorized usage of your internal network, use the following features:

- Malicious vs. compromised vs. apt zero day vs. known threats. This is enabled by association of flows to users and flows to internal organizations.
- Searching by the username will reveal accesses to different orgs and for which Apps.
- For OpenVPN users, the external IP of the user is also shown when the IP is from a geographical location different from expected. This may indicate a compromised account, especially in combination with access at odd hours.
- The OpenVPN server records logins with IP addresses and computer type, assigns IP addresses inside the lab, and sends syslog on OpenVPN.

- Use the DMF Recorder Node to retrieve the original packets for forensic analysis and to obtain evidence of unauthorized activity.

6.5 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`**. You will notice .pem files have been generated 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 7

Monitoring Network Performance and Events

This chapter describes how to monitor network performance and identify unusual events. It includes the following sections.

7.1 Interfaces Sending or Receiving Traffic

To identify specific interfaces that are sending or receiving traffic, you can use the following features:

- DMF Top Filter interfaces

- Production interfaces

Figure 7-1: DMF Filter Interfaces

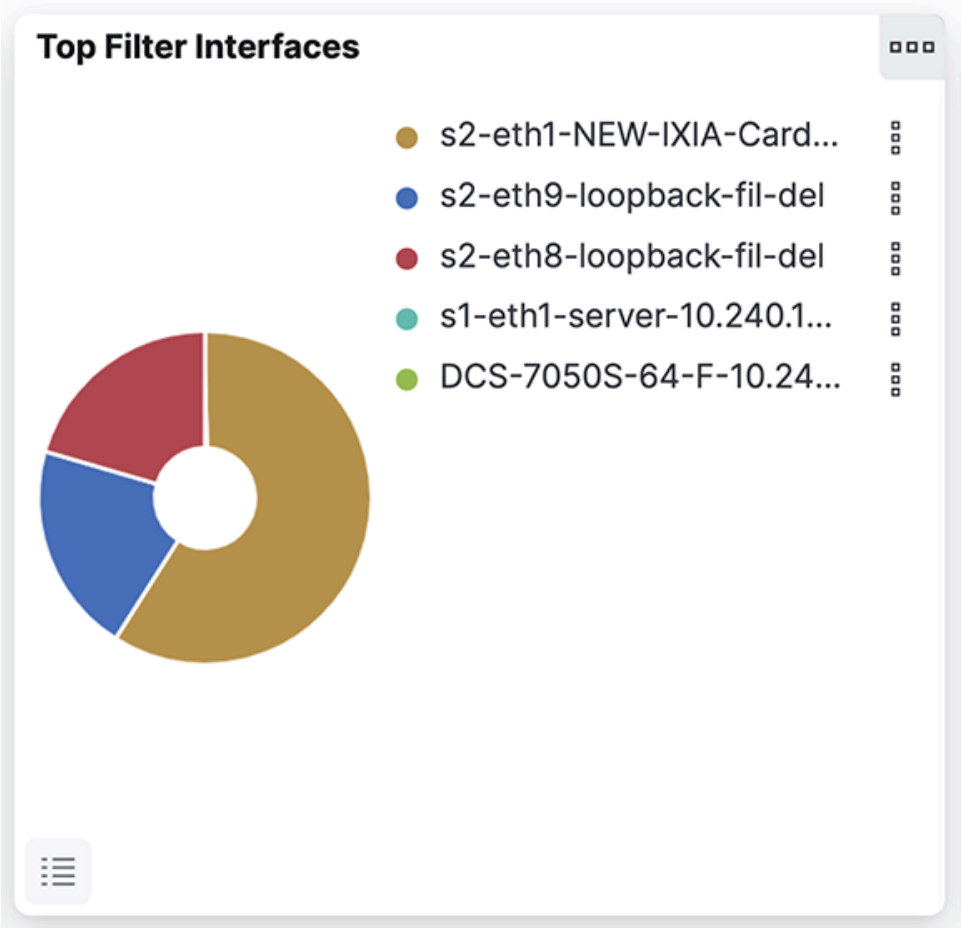
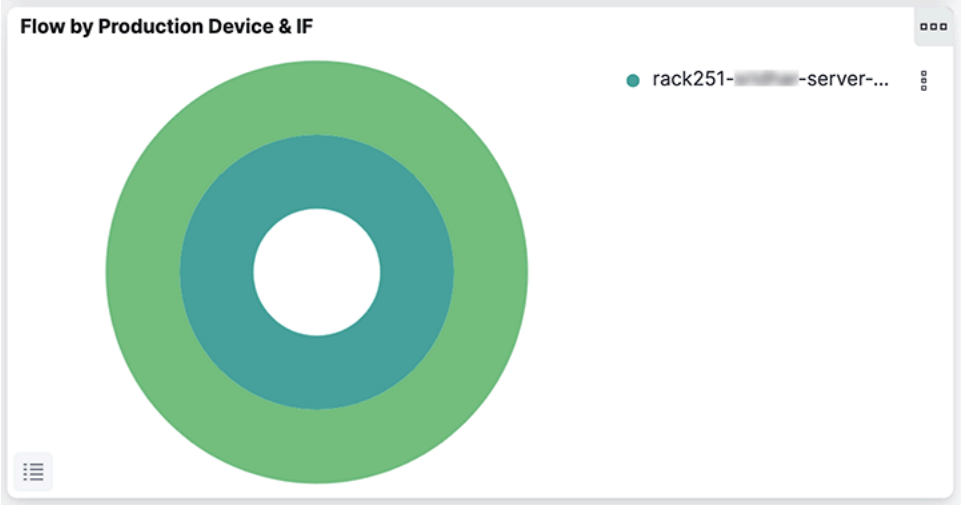


Figure 7-2: sFlow > Flow by Production Device & IF



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

7.2 Anomalies

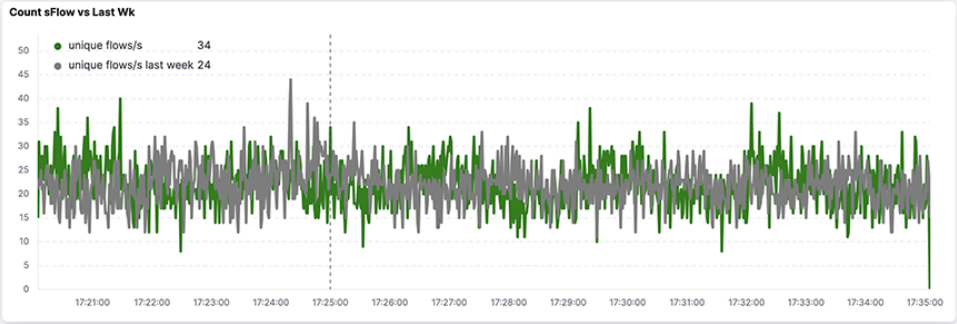
Use the following features to recognize unusual activity or events on the network.

- 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. For example, the bar visualization of traffic over time shows changing ratios of internal to external traffic, which can highlight an abnormality.

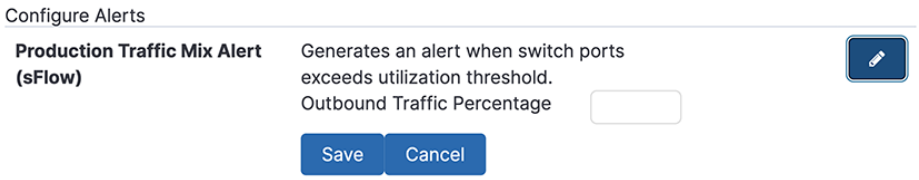
The **Count sFlow vs Last Wk** visualization in the **sFlow** dashboard shows the number of unique flows being seen now compared to last week. This visualization indicates unusual network activity and will help pinpoint a Denial of Service (DOS) attack.

Figure 7-3: Count sFlow vs Last Wk



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

Figure 7-4: Production Traffic

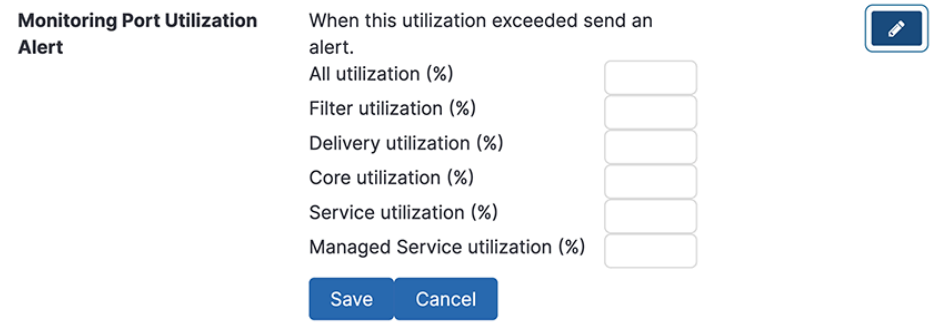


Configure utilization alerts associated with the following DMF port types:

- Filter
- Delivery
- Core

- Services

Figure 7-5: Monitoring Port Utilization Alerts



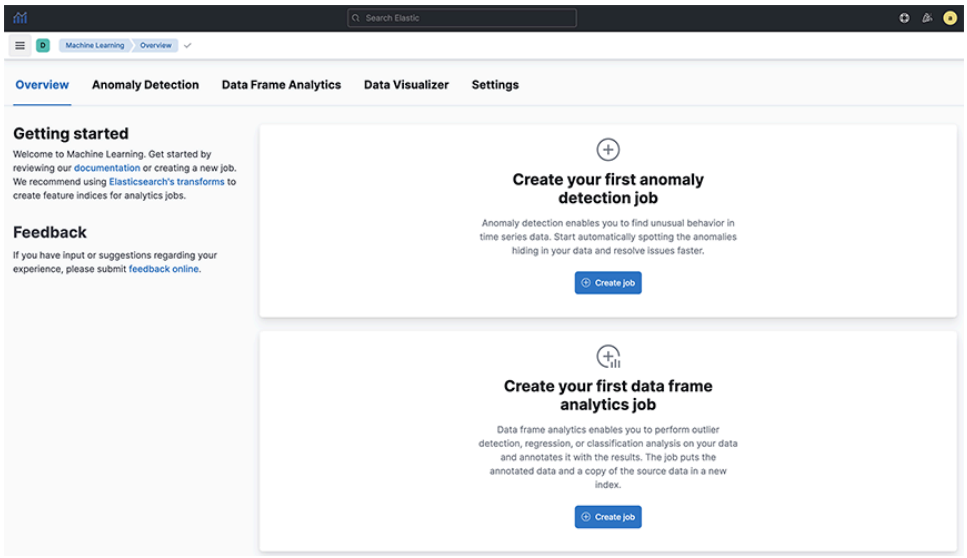
The other alerts available include the following.

- Percentage of outbound traffic exceeds usual thresholds.
- New Hosts appearing on the network every 24 hours.

Figure 7-6: New Host Report

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

Figure 7-7: Machine Learning



7.3 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.

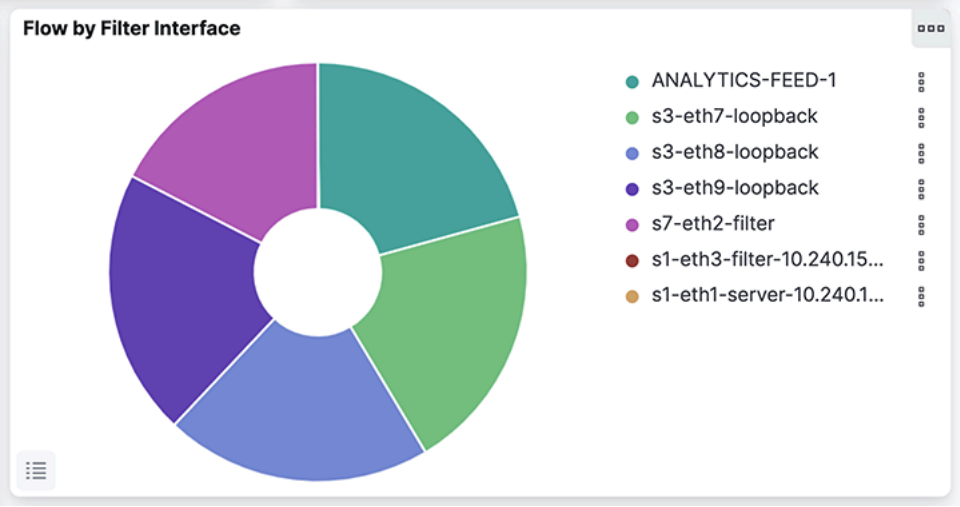
7.4 WAN Link Optimization

You can use your knowledge of DMF filter or delivery interface names to monitor traffic to or from specific interfaces. DMF WAN interface names identified with a common 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:

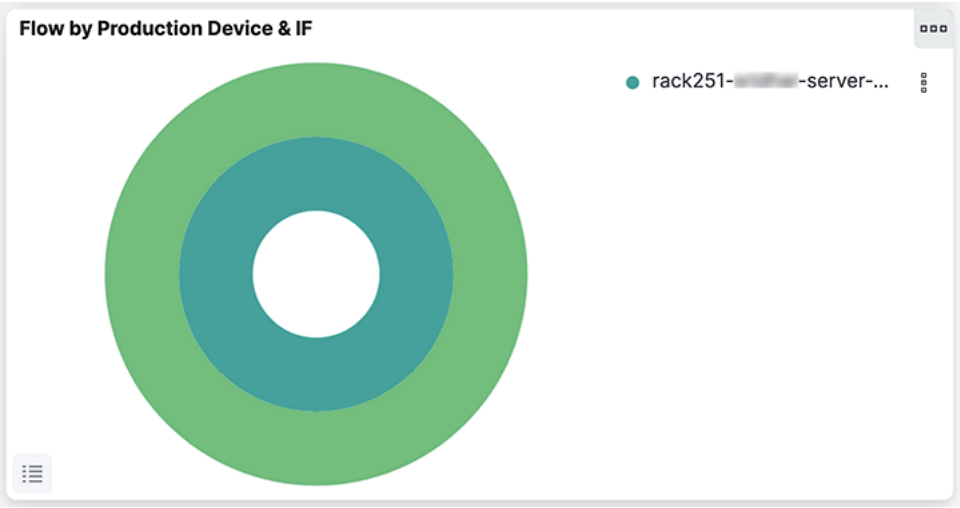
1. Select **sFlow**.
2. Refer to the **Flow by Filter Interface** visualization.

Figure 7-8: Flow by Filter Interface



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

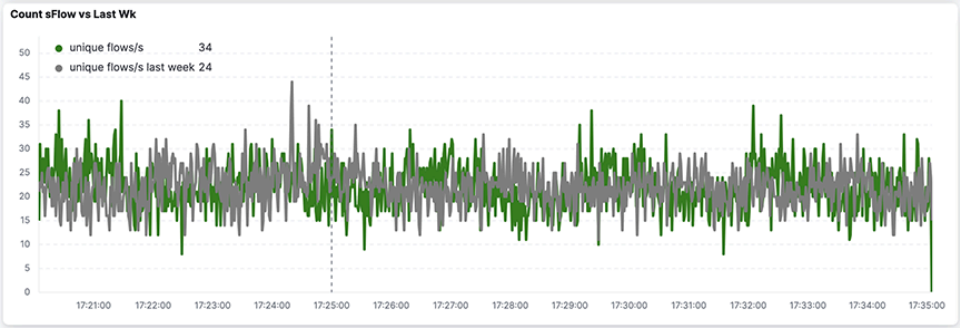
Figure 7-9: Flow by Production Device & IF



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

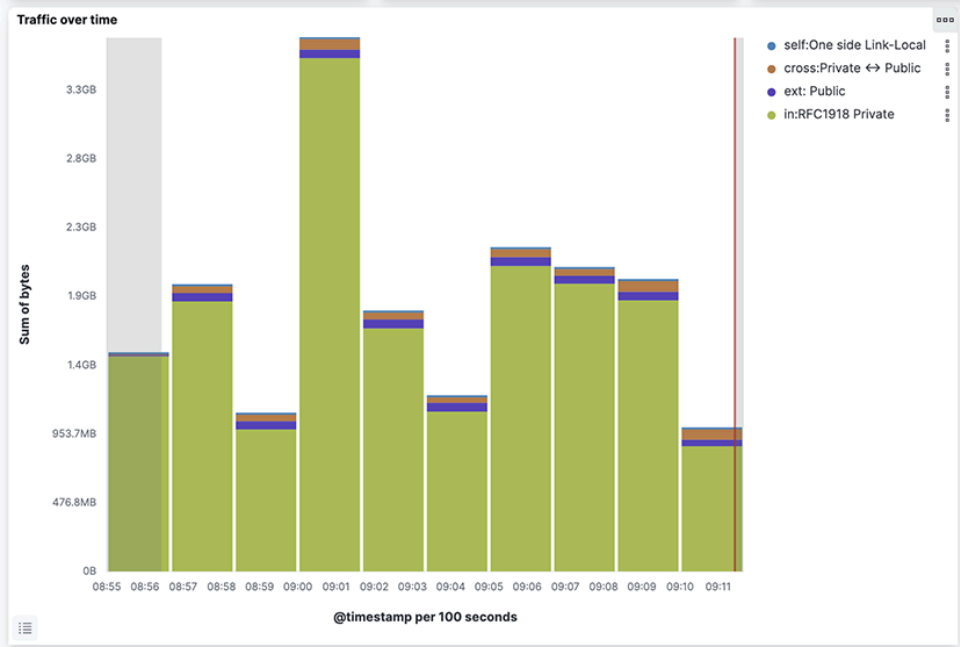
You can now refer to the **Count sFlow vs Last Wk** visualization to determine if any significant change in utilization has occurred.

Figure 7-10: Count sFlow vs Last Wk



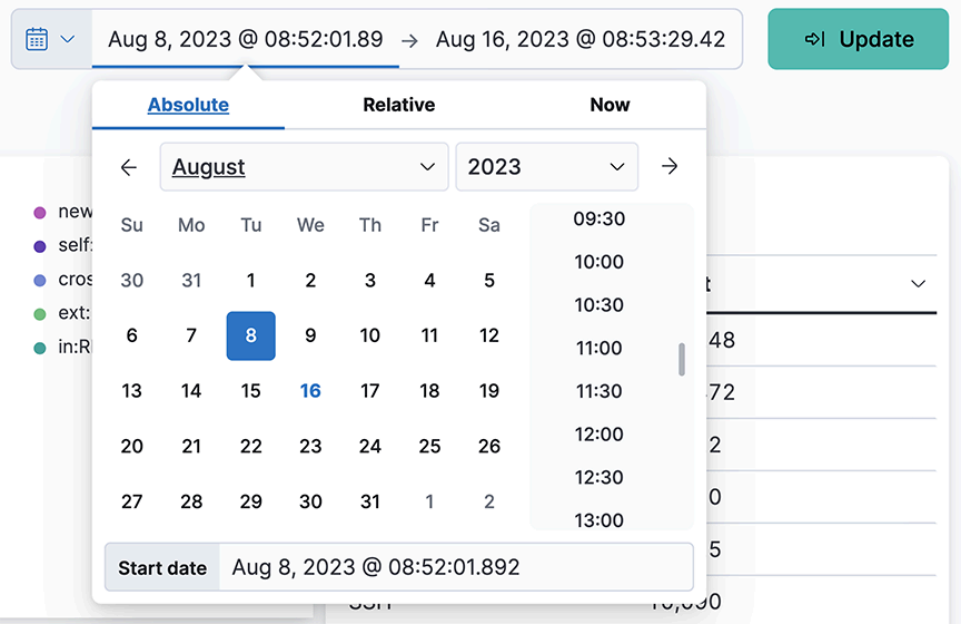
You can then use the **Traffic over Time** visualization to focus on peak and non-peak utilization periods. Drag the cursor horizontally over a peak utilization period, and the display is updated to zoom in on those events.

Figure 7-11: Traffic Over Time



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

Figure 7-12: Expanding Time Period Using the Time Range

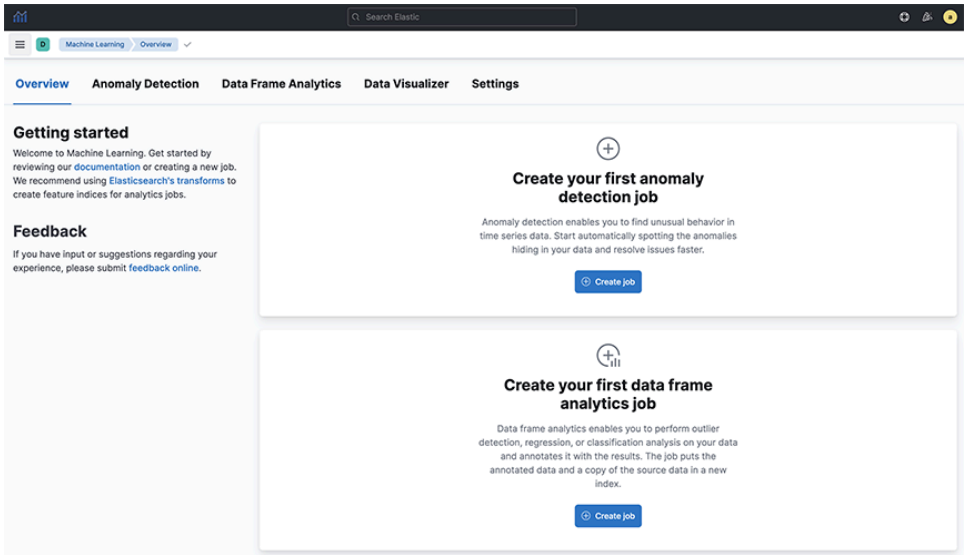


7.5 Machine Learning

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

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

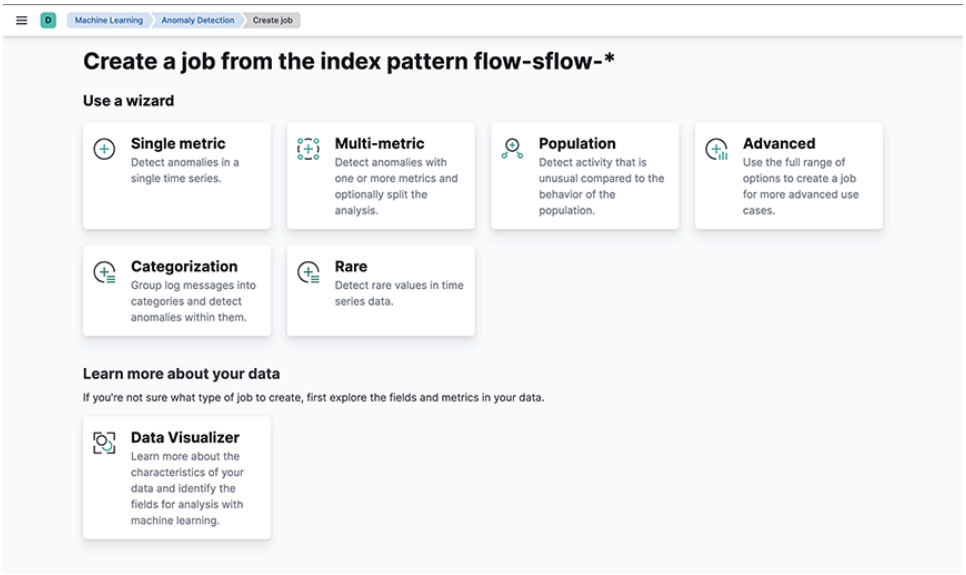
Figure 7-13: Machine Learning



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

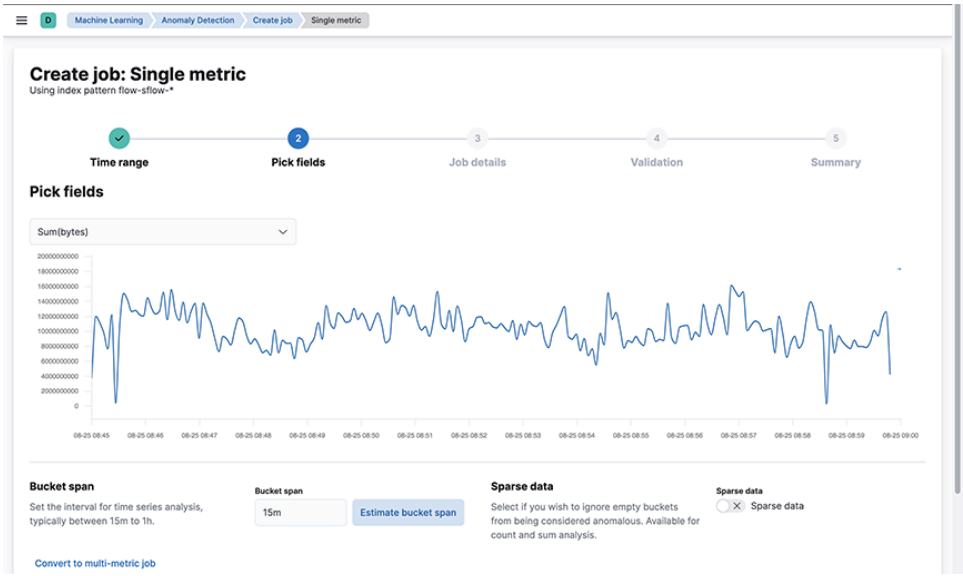
- Select the time range
- Select the appropriate metric
- Enter details: job ID, description, custom URLs, and calendars to exclude planned outages from the job

Figure 7-14: Machine Learning Job options



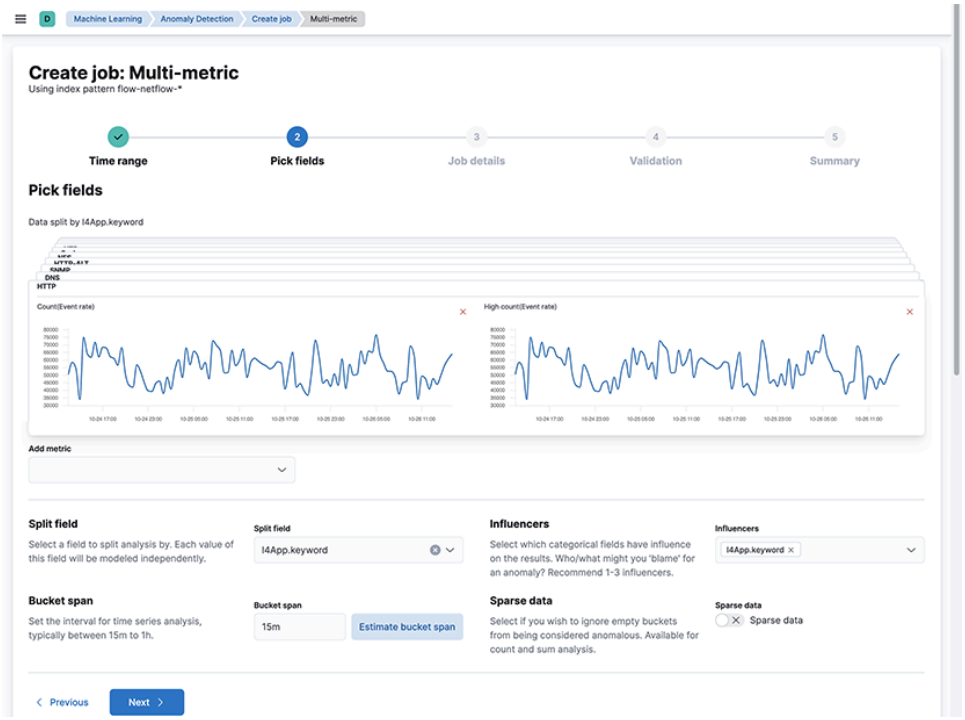
Single-metric anomaly detection uses machine learning on only one metric or field.

Figure 7-15: Single-metric Anomaly Detection



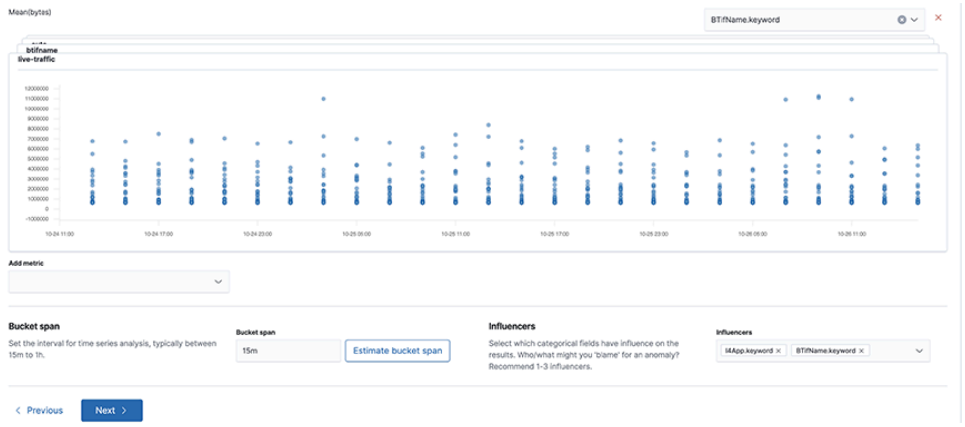
Multimetric anomaly detection uses machine learning on more than one metric field. In the image below, we are using two metrics over and running ml per l4 app.

Figure 7-16: Multimetric Anomaly Detection



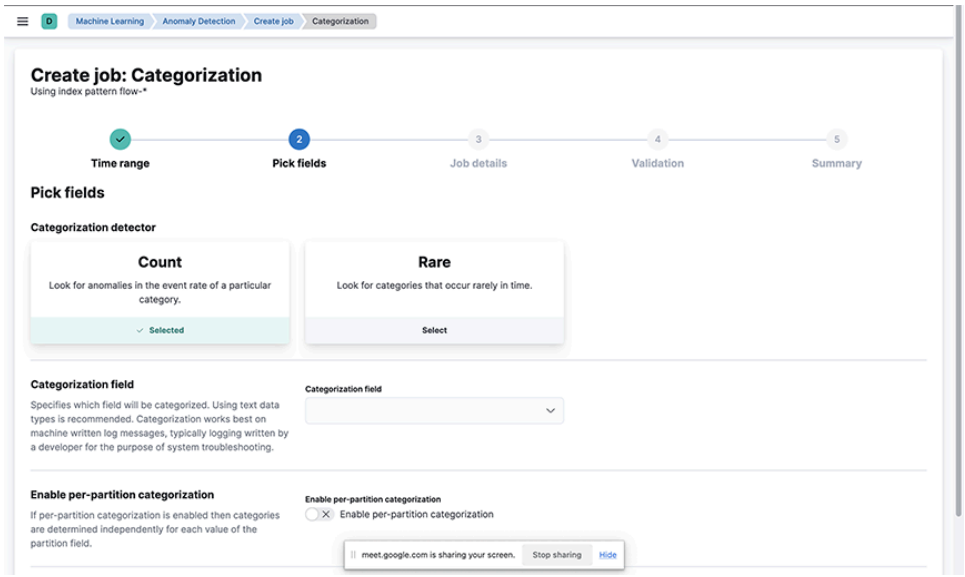
Multimetric Anomaly Detection detects network activity that differs from the population of data points. Arista Networks recommends this analysis for high-cardinality data.

Figure 7-17: Population



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

Figure 7-18: Categorization



Chapter 8

Backup and Restore

This chapter includes the following sections.

8.1 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/sn
apshot"}}' \
-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_glob
al_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_unava
ilable": 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
```

8.2 Import and Export of Saved Objects

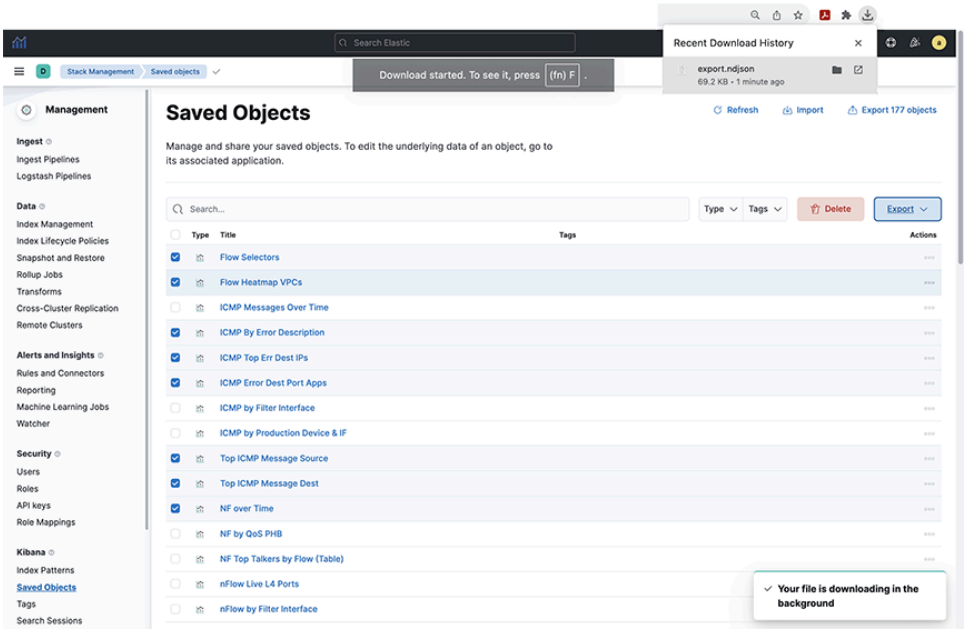
The **Saved Objects** UI helps keep track of and manage saved objects. These objects store data for later use, including dashboards, visualization, searches, and more. This section explains the procedures for backing up and restoring saved objects in Arista Analytics.

8.2.1 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 **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 8-1: Verifying a Saved/Downloaded Object



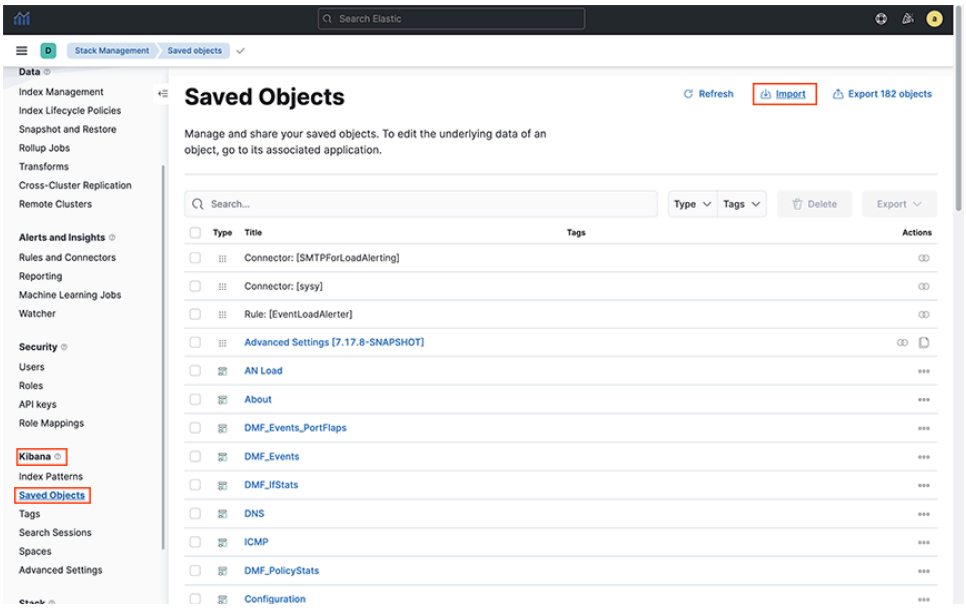
Note: Recommended Best Practices

- While creating saved objects, Arista Networks recommends using the naming convention that suits your environment. For instance, in the example above, 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 dashboards individually based on their type is a more appropriate option, as tracking modifications to a dashboard improves using this method. Dashboards should use 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.

8.2.2 Importing Saved Objects

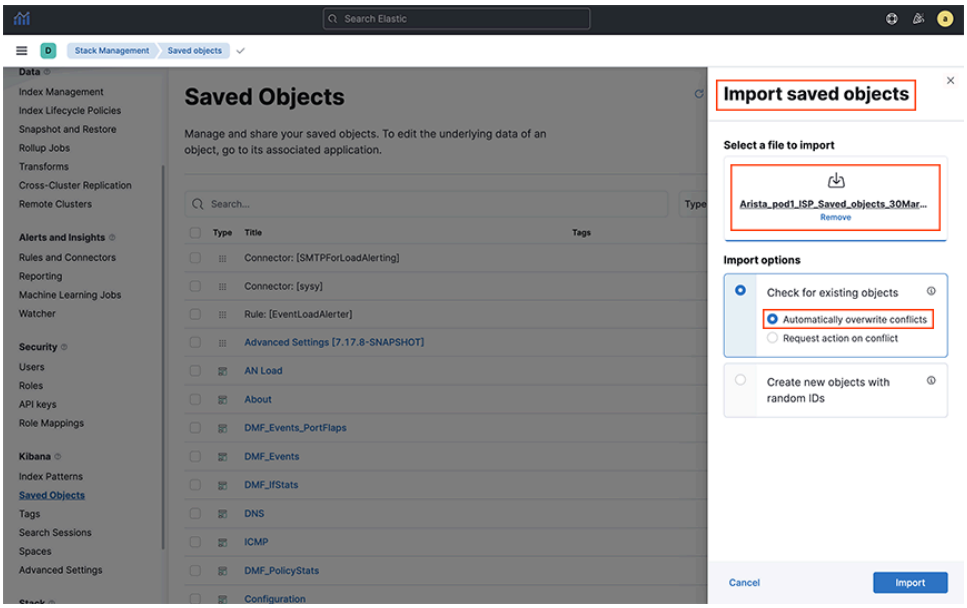
1. To import one or a group of custom-created objects, navigate to **Main Menu** > **Management** > **Kibana** > **Saved Objects**.

Figure 8-2: Importing a Group of Saved Objects



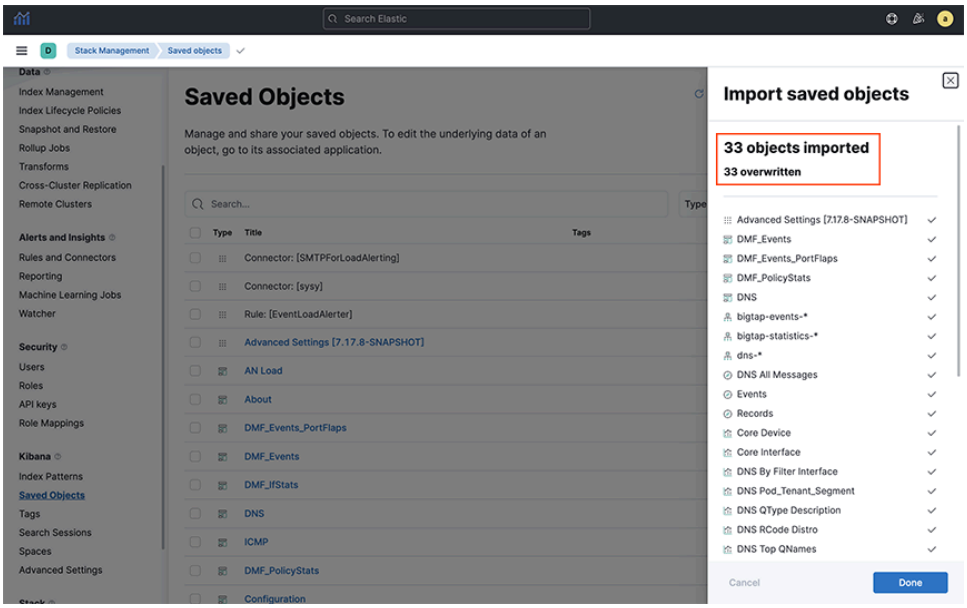
2. Click **Import** and navigate to the NDJSON file that represents the objects to import. By default, saved objects already in Kibana are overwritten by the imported object. The system should display the following screen.

Figure 8-3: 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 object**.

Figure 8-4: Import Successful Dialog Box



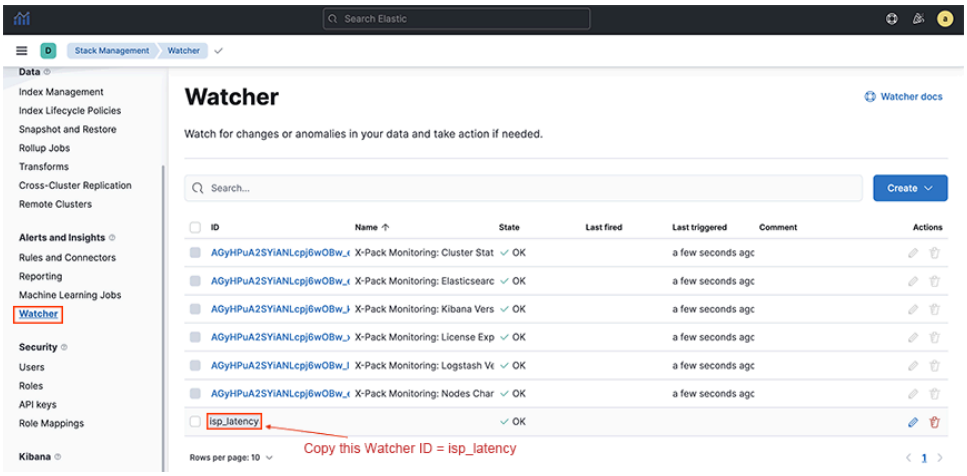
8.3 Import and Export of Watchers

Use the Watcher feature to create actions and alerts based on certain conditions and periodically evaluate them using queries on the data. This section explains the procedure of backing up and restoring the watchers in Arista Analytics.

8.3.1 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 8-5: Find Watcher_ID



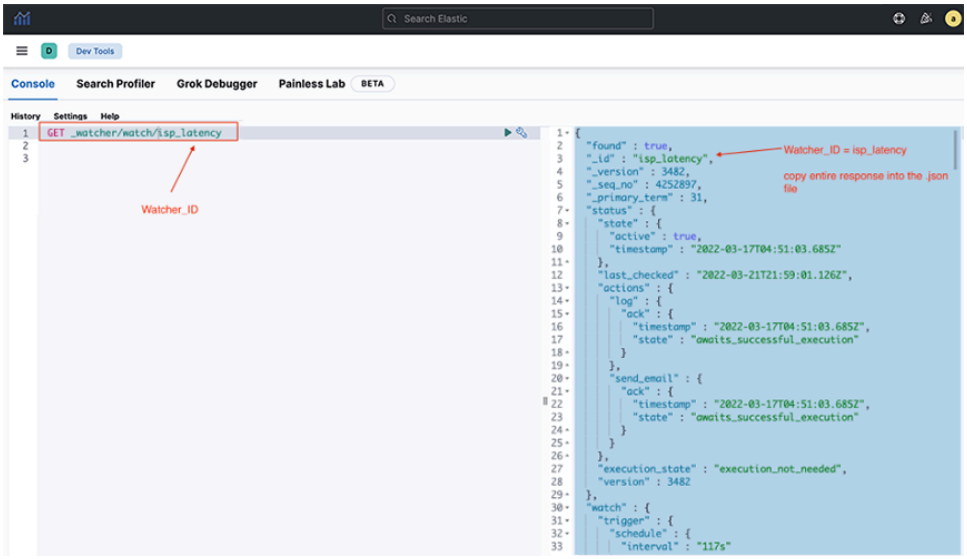
2. Open the main menu, then select **Dev Tools > Console**. Issue the **`GET`** API mentioned below with the **`watcher_id`**. The response appears in the output terminal.

   Run the following API call:

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

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

   Figure 8-6: 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 useful nomenclature: **`Arista_pod1_2022-02-03_isp_latency_watcher.json`**.

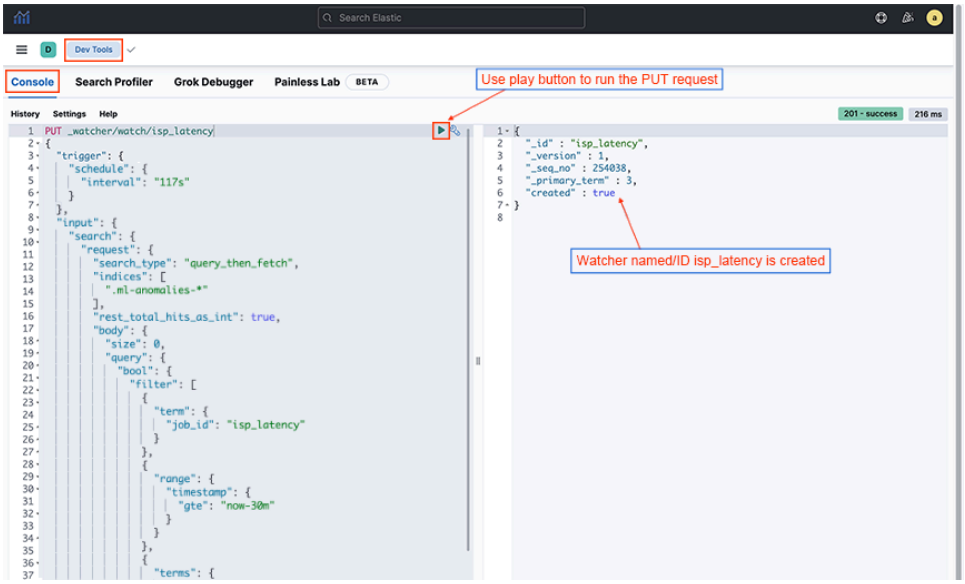
8.3.2 Importing Watchers

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

Figure 8-7: jq Command Output

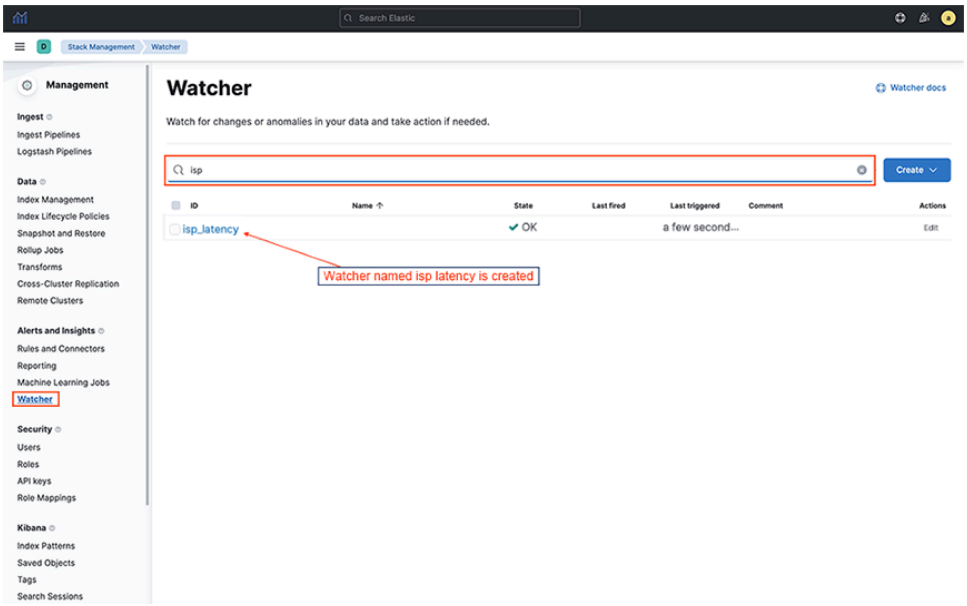
2. Click **DevTools** > **console**, enter the API **`PUT_watcher/watch/<watcher_id>`**, and copy the **Step 1** output into the screen shown below. Replace ***watcher_id*** with the desired watcher name. The output terminal will confirm the creation of the watcher.

Figure 8-8: PUT API in Dev Tools Console



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

Figure 8-9: Watcher



8.4 Import and Export of Machine Learning Jobs

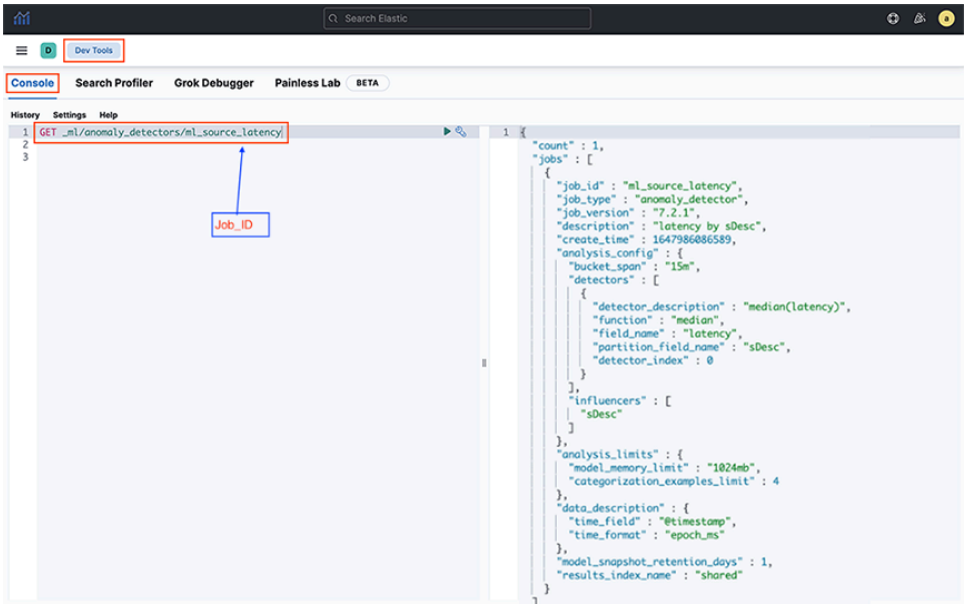
Machine Learning (ML) automates time series data analysis by creating accurate baselines of normal behavior and identifying anomalous patterns. This section explains how to back up and restore the machine learning jobs in Arista Analytics.

8.4.1 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 8-10: Main Menu > Dev Tools > Console



2. Copy the **`GET`** API response of the ML job into a **`.json`** file with nomenclature 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`**

8.4.2 Importing Machine Learning Jobs

1. Not all exported fields must be imported. 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 8-11: jq Required Fields

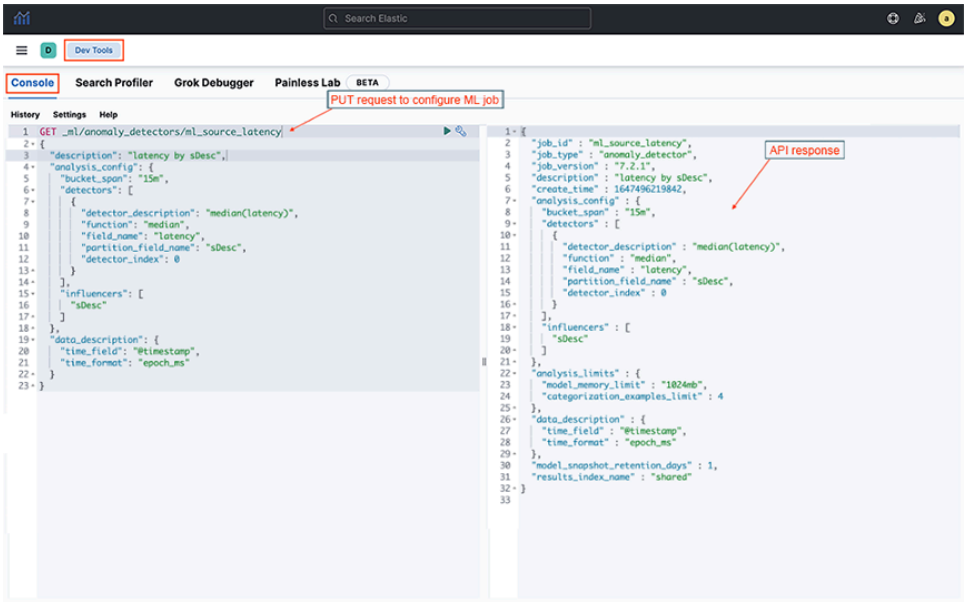
2. Select **Dev tools** > **Console** and copy the **Step 1** output into the screen shown below along with the **`PUT`** request.

Run the following API call:

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

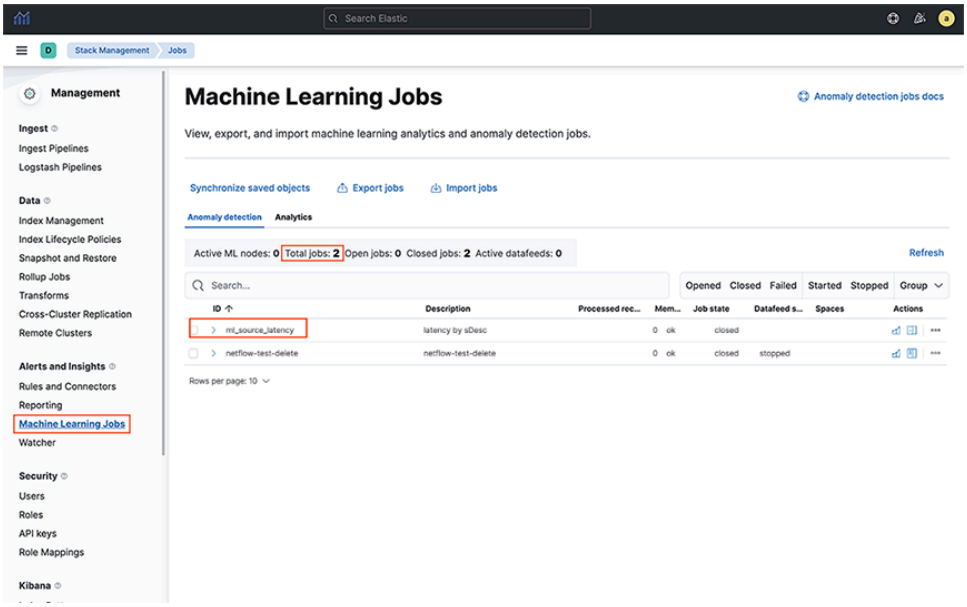
Replace ***ml_job_name*** with the specific string of the ML Job name.

Figure 8-12: 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 8-13: ML Job Verification



Chapter 9

Using TACACS+ and RADIUS to Control Access to the Arista Analytics CLI

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

9.1 Using AAA Services with Arista Analytics

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

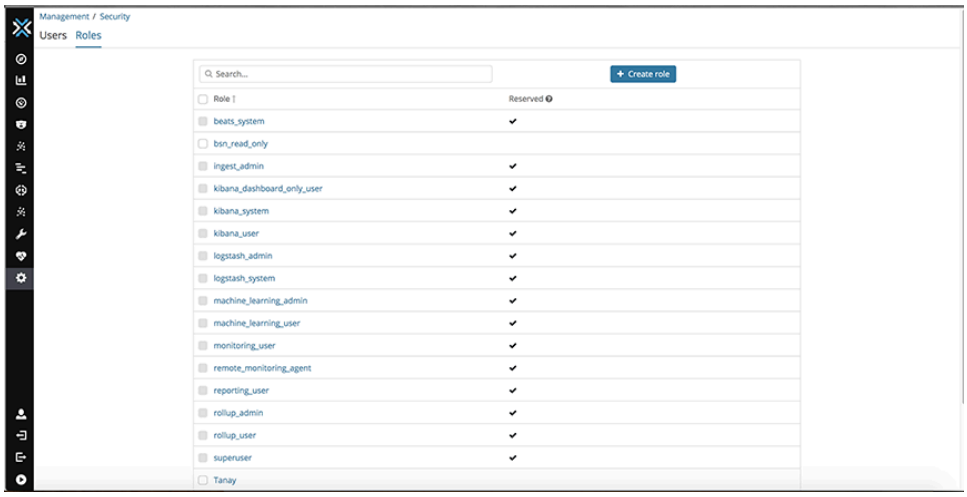
The following table lists the accepted Attribute-Value (AV) pairs:

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 allow creation of 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 that is configured in the CLI.

Figure 9-1: Creating a Group in Elasticsearch



A remotely authenticated admin user has full administrative privileges. Read-only users on the switch must be remotely authenticated. 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 is 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

You can use a TACACS+ server to maintain administrative access control instead of using the Analytics Node local database, although it is a best practice to maintain the local database as the secondary method of authentication and authorization in case the remove server becomes unavailable.

9.1.1 DMF TACACS+ Configuration

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

Authentication Method

- Configure the TACACS+ server to accept ASCII authentication packets. Do not use 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: To use the same user credentials to access ANET and non-ANET devices, the BSN-User-Role attribute must be specified as **Optional** in the **`tac_plus.conf`** file.

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
```

Now, 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 a 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)]
```

9.2 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)#
```

The TACACS+ key value is displayed as a type7 secret instead of plaintext.

To configure the Analytics Node with TACACS+ to control administrative access to the switch, complete the following steps.

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, the following command enables TACACS+ with the server running at ***10.2.3.4***, using a plaintext key.

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

If the key is omitted, an empty key is used.

Note: Do not use the pound character (#) in the TACACS secret. It will be interpreted as 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 individual 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`**.

If no key is specified for a given host, then the global key value is used. If no key is specified globally and no key is specified for a given host, then an empty key is assumed.

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 when you configure TACACS+ to avoid disabling access to the Analytics Node.

9.3 Setting up a TACACS+ Server

For further details, or for instructions for setting up other servers, refer to your AAA server documentation.

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. The user account will be checked locally only when the remote server is unreachable. In this example, accounting is set to store audit logs locally and send them to the remote server.

9.3.1 Using the Same Credentials for the Analytics Node and Other Devices

To use the same user credentials to access both the Analytics Node and other devices, the ***BSN-User-Role*** attribute must be specified as **Optional** in the **`tac_plus.conf`** file, as shown in the following example.

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

9.3.2 RBAC-based Configuration for Non-default Group User

To create an RBAC configuration for a user in a non-default group, complete the following steps:

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

```
group AD1
```

   Do not associate 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. s

9.4 Using RADIUS for Managing Access to the Arista Analytics Node

Note: RADIUS does not separate authentication and authorization, so be careful when authorizing a user account with a remote RADIUS server to use the password that is 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: The **admin** and **recovery** user accounts cannot be authenticated remotely using TACACS. These accounts are always authenticated locally to prevent administrative access from being lost in case a remote AAA server is not available.

The **admin** group provides full 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 is used for sending CLI command activity accounting to the RADIUS server.

9.4.1 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 in which they are configured.

9.4.2 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: This is intended only to show how the VSA is associated with the user and its privileges. In an actual deployment, a database and encrypted password are necessary.

   The following example authorizes ***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 is set to store audit logs locally and send them to the remote server.

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

   This is required if 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

Creating Watcher Alerts for Machine Learning jobs

This appendix describes how to create watcher alerts for machine learning jobs.

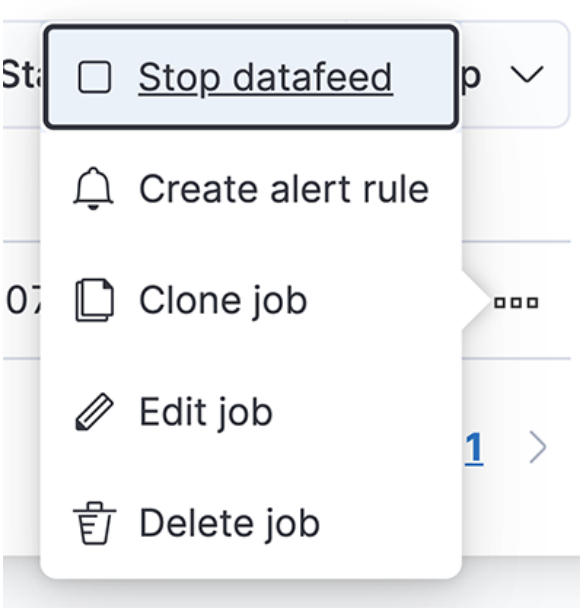
A.1 Watcher Alert Workaround

DMF 8.1 uses ***Elasticsearch 7.2.0*** where the inter-container functional calls were HTTP-based. However, ***DMF 8.3*** uses ***Elasticsearch version 7.13.0*** where these calls are now required to be HTTPS-based. This would require an extensive change in the system calls used by the Analytics Node and this effort is being worked on by engineering. Arista recommends the below workaround until the above fixes are released.

Workaround Summary: Create a Watcher manually using the provided template. Configure the watcher to use the job id for the ML job that needs to send alerts. Use 'webhook' as the alerting mechanism within the Watcher to send alerts to 3rd party tools like 'Slack'.

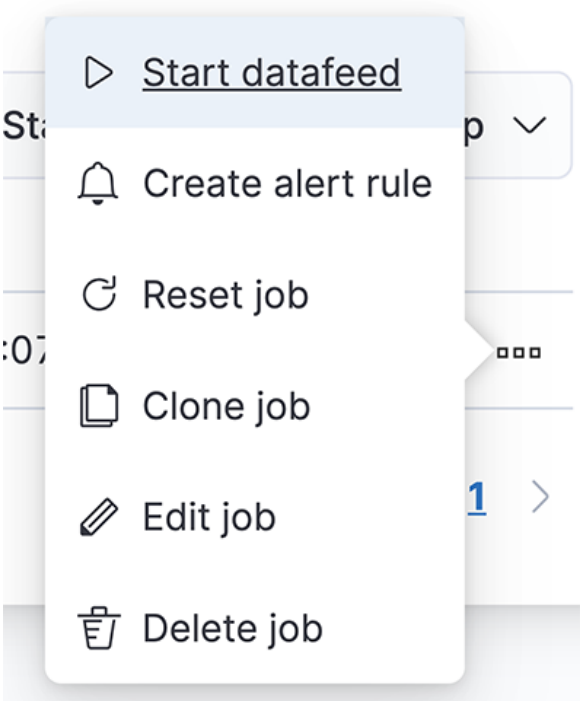
1. Access the ML job page on the AN and then click **Manage Jobs** to list the ML jobs.
2. If the data feed column shows as **stopped**, skip to **Step 3**. If it says **started**, click the **3 dots** for a particular ML job and **Stop** the data feed for the current ML job.

Figure A-1: Stop Data Feed



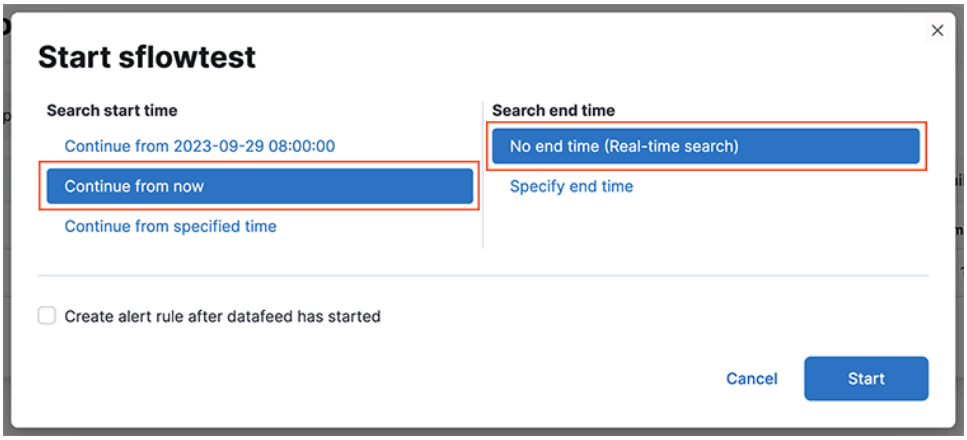
3. After the data feed has stopped, click the **3 dots** and start the data feed.

Figure A-2: Start Data Feed



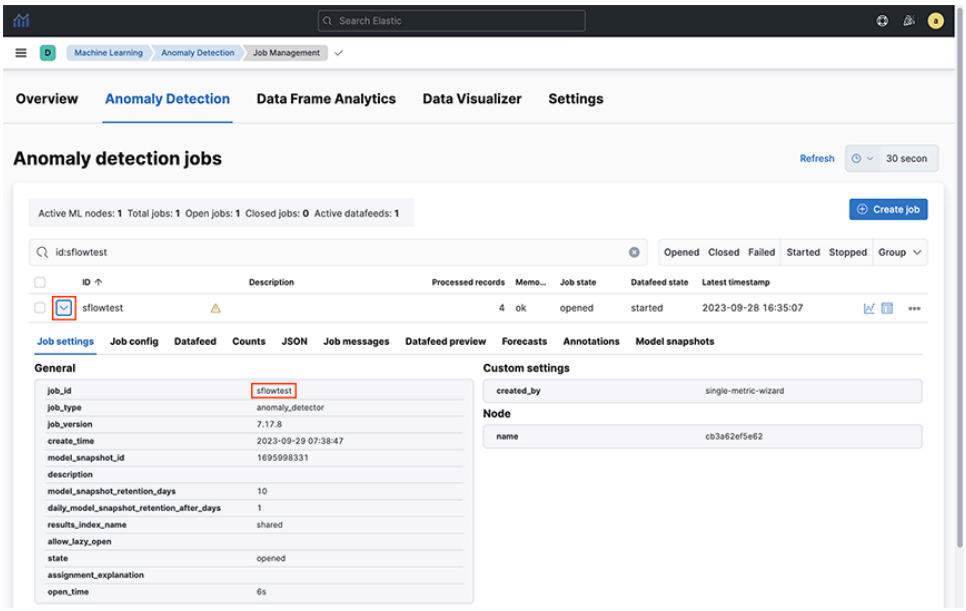
4. Select the options as shown in the diagram below.

Figure A-3: Job Time Options



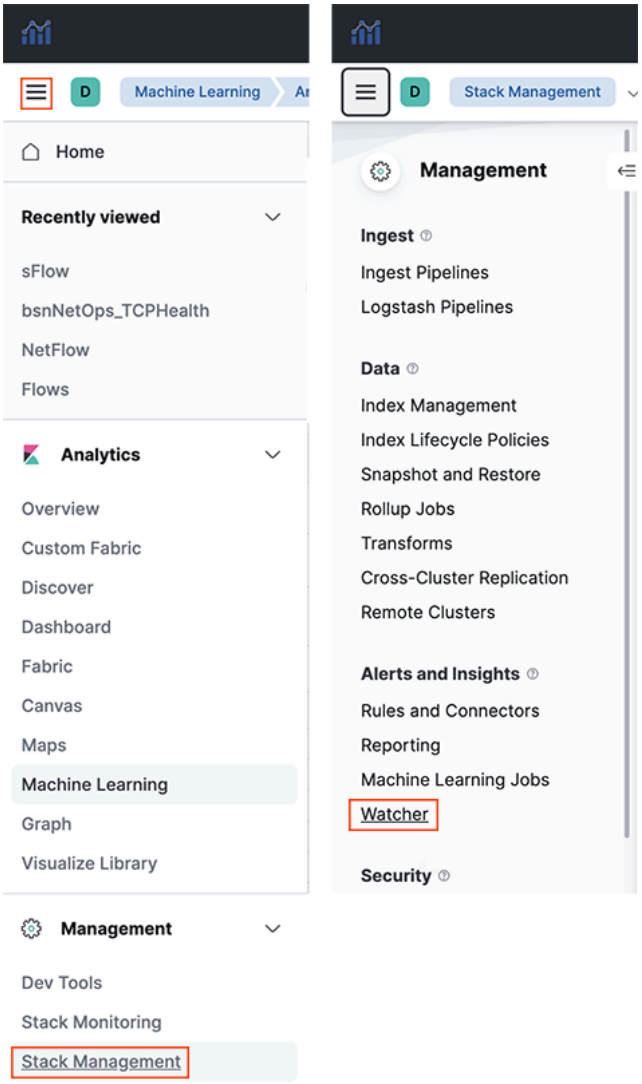
5. Confirm that the data feed has started. Note down the job id of this ML job.

Figure A-4: ML Job Characteristics



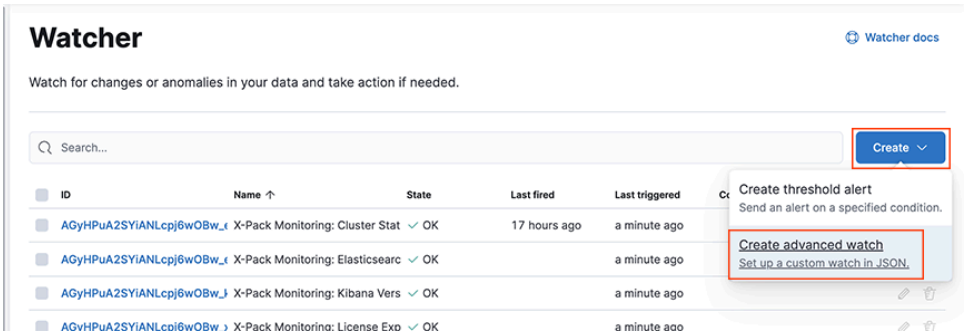
6. Access the **Watchers** page.

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.

Appendix B

References

B.1 Related Documents

The following documentation is available for ***Arista Analytics 8.5.0***:

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