User Guide

DANZ Monitoring Fabric

Version 8.8

Arista.com

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

© Copyright 2025 Arista Networks, Inc. All rights reserved. The information contained herein is subject to change without notice. The trademarks, logos, and service marks ("Marks") displayed in this documentation are the property of Arista Networks in the United States and other countries. Use of the Marks is subject to the Arista Networks Terms of Use Policy, available at www.arista.com/en/terms-of-use. Use of marks belonging to other parties is for informational purposes only.

Contents

Chapter 1: Introduction to DMF and Overview..... 1
1.1 Out-of-band Monitoring with DANZ Monitoring Fabric..... 2
1.2 Using the DANZ Monitoring Fabric GUI..... 4
1.2.1 Admin Settings..... 7
1.2.2 DMF Features Page..... 12
1.2.3 Dashboard Layout..... 28
1.2.4 Controller Health..... 29
1.2.5 Switch Health..... 35
1.2.6 Policy Health..... 45
1.2.7 Smart Node Health..... 48
1.2.8 Empty State..... 53
1.2.9 Alerts..... 56
1.3 Using the Command Line Interface..... 63

Chapter 2: Managing DMF Switches and Interfaces..... 66
2.1 DMF Switches Dashboard..... 67
2.1.1 Overview..... 68
2.1.2 Inventory..... 70
2.1.3 Configuration..... 76
2.1.4 Properties..... 83
2.1.5 IP Address Allocation..... 85
2.1.6 TCAM Utilization..... 87
2.1.7 Provision Switch..... 90
2.1.8 Firmware..... 92
2.1.9 Revert to the Legacy Dashboard..... 96
2.2 Accept Additional MAC Address Formats..... 96
2.3 DMF Interfaces..... 98
2.3.1 Overview..... 99
2.3.2 Configuration..... 102
2.3.3 Operational..... 116
2.3.4 Traffic Counters..... 120
2.3.5 Configure a DMF Interface..... 122
2.3.6 Identify a Filter Interface using Destination MAC Rewrite..... 130
2.3.7 Configure sFlow® on a Filter and Delivery Interface..... 134
2.3.8 Configure Error Disable State..... 137
2.3.9 Configure Interface Groups..... 141
2.3.10 Using the Command Line Interface..... 150
2.4 Switch Light CLI Operational Commands..... 167

Chapter 3: Managing DMF Policies..... 172
3.1 Overview..... 173
3.2 DMF Policies Page..... 174
3.3 Policy Elements..... 194
3.4 Configure a Policy..... 196
3.4.1 Create a New Policy..... 198
3.4.2 Traffic Sources..... 202
3.4.3 Services..... 211

3.4.4 Destination Tools..... 214
3.4.5 Stat Picker..... 224
3.4.6 Match Traffic and Match Traffic Rules..... 226
3.4.7 Rules Table..... 236
3.4.8 Persist UDF Input Format..... 239
3.5 Using the Packet Capture Action in a Policy..... 241
3.6 Using the Command Line Interface..... 243
3.6.1 Configure a Policy..... 244
3.6.2 Stop, Start, and Schedule a Policy..... 246
3.6.3 Clear a Policy..... 248
3.6.4 View Policies..... 249
3.6.5 Rename a Policy..... 250
3.6.6 Define Out-of-band Match Rules..... 251
3.6.7 Define a Policy with a Packet Capture Action..... 252
3.6.8 Rename Object..... 254
3.6.9 Persist UDF Input Format..... 256

Chapter 4: Monitoring Fabric and Production Networks..... 257
4.1 Monitor DMF Interfaces..... 258
4.2 Viewing Oversubscription Statistics..... 259
4.3 View Fabric-Connected Devices..... 260
4.4 Using the Command Line Interface..... 262
4.4.1 Monitor Interface Configuration..... 263
4.4.2 View Switch Configuration..... 265
4.4.3 View Connected Devices and LAGs..... 266
4.4.4 Viewing Information about a Connected Production Network..... 267

Chapter 5: DMF Service Node Appliance..... 268
5.1 Overview..... 269
5.2 Changing the Service Node Default Configuration..... 270
5.3 Using SNMP to Monitor DPDK Service Node Interfaces..... 271
5.4 Configuring Managed Services..... 272
5.4.1 Using the GUI to Define a Managed Service..... 273
5.4.2 Using the CLI to Define a Managed Service..... 275
5.4.3 Monitoring Managed Services..... 277
5.5 Deduplication Action..... 278
5.6 Header Strip Action..... 280
5.6.1 Configuring the Post-service Match..... 283
5.7 IPFIX and Netflow Actions..... 284
5.7.1 IPFIX Template..... 285
5.7.2 IPFIX Keys..... 286
5.7.3 IPFIX Fields..... 287
5.7.4 Active and Inactive Timers..... 288
5.7.5 Example Flowset..... 289
5.7.6 Using the GUI to Define an IPFIX Template..... 290
5.7.7 Using the CLI to Define an IPFIX Template..... 293
5.7.8 Using the GUI to Define an IPFIX Service Action..... 295
5.7.9 Using the CLI to Define an IPFIX Service Action..... 296
5.7.10 Records Per Interface Netflow using DST-MAC Rewrite..... 297
5.8 Packet-masking Action..... 299
5.9 Arista Analytics Node Capability..... 300
5.9.1 Configuring the Arista Analytics Node Using the GUI..... 301
5.9.2 Configuring the Arista Analytics Node Using the CLI..... 302
5.10 Pattern-drop Action..... 304

5.11 Pattern-match Action..... 305
5.12 Slice Action..... 307
5.13 Packet Slicing on the 7280 Switch..... 309
5.13.1 Using the CLI to Configure Packet Slicing - 7280 Switch..... 310
5.13.2 Using the GUI to Configure Packet Slicing - 7820 Switch..... 312
5.13.3 Troubleshooting Packet Slicing..... 319
5.13.4 Packet Slicing Considerations..... 320
5.14 VXLAN Stripping on the 7280R3 Switch..... 321
5.14.1 Using the CLI to Configure VXLAN Header Stripping..... 322
5.14.2 Using the GUI to Configure VXLAN Header Stripping..... 324
5.14.3 Syslog Messages..... 329
5.14.4 Troubleshooting..... 329
5.14.5 Limitations..... 329
5.15 Session Slicing for TCP and UDP Sessions..... 330
5.15.1 Using the CLI to Configure Session Slicing..... 331
5.15.2 Using the GUI to Configure Session Slicing..... 332
5.16 Timestamp Action..... 337
5.17 UDP-Replication Action..... 338
5.18 Redundancy of Managed Services in Same DMF Policy..... 340
5.18.1 Using the GUI to Configure a Backup Managed Service..... 341
5.18.2 Using the CLI to Configure a Backup Managed Service..... 342
5.19 Application Identification..... 343
5.19.1 Using the CLI to Configure Application Identification..... 344
5.19.2 Using the CLI to Configure app-id..... 345
5.19.3 Using the CLI to Configure app-id-filter..... 346
5.19.4 Using the CLI to Configure app-id and app-id-filter Combined..... 348
5.19.5 Using the GUI to Configure app-id and app-id-filter..... 349
5.19.6 Dynamic Signature Updates (Beta Version)..... 355
5.19.7 CLI Show Commands..... 357
5.19.8 Syslog Messages..... 359
5.19.9 Scale..... 360
5.19.10 Troubleshooting..... 361
5.19.11 Limitations..... 362
5.20 Redundancy of Managed Services Using Two DMF Policies..... 363
5.21 Sharing Managed Services Across Policies..... 364
5.21.1 Configuration using the CLI..... 365
5.21.2 Show Commands..... 367
5.21.3 Troubleshooting..... 377
5.21.4 Limitations..... 377
5.22 Cloud Services Filtering..... 378
5.22.1 Cloud Services Filtering Configuration..... 379
5.23 Multiple Services Per Service Node Interface..... 380
5.24 Sample Service..... 381
5.24.1 Sample Service CLI Configuration..... 382
5.24.2 Sample Service GUI Configuration..... 383
5.24.3 Troubleshooting Sample Service..... 385
5.24.4 Limitations..... 386
5.25 Flow Diff Latency and Drop Analysis..... 387
5.25.1 Configure Flow Diff Latency and Drop Analysis Using the CLI..... 388
5.25.2 Configuring Tap Points..... 390
5.25.3 Configuring Policy..... 391
5.25.4 Show Commands..... 393
5.25.5 Syslog Messages..... 395
5.25.6 Troubleshooting..... 396
5.25.7 Limitations..... 398
5.25.8 Resources..... 400

5.26 Sharing L3 Delivery Interfaces Across Services..... 401
5.27 Service Node Management Migration L3ZTN..... 405

Chapter 6: DMF Recorder Node..... 406

6.1 Overview..... 407
6.2 Configuration Summary..... 408
6.3 Indexing Configuration..... 409
6.4 Pre-buffer Configuration and Events..... 411
6.5 Using an Authentication Token..... 412
6.6 Recorder Node User Interface..... 413
6.6.1 Features and Functions..... 414
6.6.2 Onboard a Recorder Node..... 422
6.6.3 Configure a Recorder Node Interface..... 430
6.6.4 Edit a Recorder Node Interface..... 432
6.6.5 Recorder Node Details..... 434
6.6.6 Recorder Node Events..... 439
6.6.7 Assign a Recorder Node Interface to a Policy..... 445
6.6.8 Recorder Node Query..... 454
6.6.9 Manage Access to the Recorder Node..... 472
6.6.10 Enabling Egress sFlow® on Recorder Node Interfaces..... 480
6.6.11 Empty State..... 484
6.7 Using the Recorder Node Command Line Interface..... 485
6.7.1 Manage the DMF Recorder Node..... 485
6.7.2 Run Recorder Node Queries..... 495
6.7.3 Viewing Query History..... 500
6.7.4 Using RBAC to Manage Access to the DMF Recorder Node..... 501
6.7.5 View Information About a Recorder Node..... 503
6.7.6 Changing the Recorder Node Default Configuration..... 507
6.7.7 Large PCAP Queries..... 509
6.7.8 Recorder Node Management Migration L3ZTN..... 510
6.7.9 Recorder Node Show Commands..... 511
6.7.10 Ability to Deduplicate Packets - Query from Recorder Node..... 512
6.7.11 Enable Egress sFlow..... 514
6.8 Considerations and Limitations..... 514
6.8.1 Deduplication Limitations..... 515
6.8.2 Guidelines and Limitations for Enabling Egress sFlow..... 516
6.8.3 Recorder Node Recording State API Limitations..... 516
6.9 Recorder Node Support for CVaaS..... 517

Chapter 7: Link Aggregation..... 520

7.1 Configuring Link Aggregation..... 521
7.1.1 Configure Link Aggregation Groups..... 522
7.1.2 L2 GRE Key Hashing..... 529
7.1.3 VXLAN Hashing..... 531
7.1.4 MLAG RTAG7 Hash Computation for Unbalanced Load Balancing..... 533
7.1.5 Trident4 LAG Hashing for L2GRE and VXLAN Traffic..... 536
7.2 Pseudo Multi-Chassis Link Aggregation..... 537
7.3 MLAG Components..... 538
7.4 MLAG Limitations..... 539
7.5 Configure MLAG..... 540
7.6 Create MLAG Policy..... 543
7.7 Viewing Policy Statistics..... 545
7.8 Viewing MLAG Links..... 548
7.9 Using the Command Line Interface..... 549

7.9.1 Configure Link Aggregation Groups....550
7.9.2 Configuring Hashing Fields....551
7.9.3 Minimum Link and Activation Configuration for LAG Interfaces....553
7.9.4 Tunnel Endpoint for vCenter Integration....555
7.9.5 L2 GRE Key Hashing....555
7.9.6 VXLAN Hashing....556
7.9.7 VXLAN LAG Hashing on DCS7280 Platforms....558
7.9.8 MLAG RTAG7 Hash Computation for Unbalanced Load Balancing....559
7.9.9 Configuring an MLAG....561
7.9.10 MLAG Link Discovery....564
7.9.11 Overlapping Policies in LAGs....565
7.9.12 Using LAG Interfaces as Members in MLAG Interfaces....567

Chapter 8: Tunneling Between Data Centers....568
8.1 Understanding Tunneling....569
8.2 Encapsulation Type....571
8.3 Using Tunnels in Policies....572
8.4 Configure a GRE Tunnel....573
8.5 Configure a VXLAN Tunnel....575
8.6 Viewing or Modifying Existing Tunnels....577
8.7 Using a Tunnel with User-defined Offsets....578
8.8 Configuring Wildcard Tunnels....580
8.9 Using the Command Line Interface....584
8.9.1 Configure a GRE Tunnel....585
8.9.2 Configure a VXLAN Tunnel....588
8.9.3 Configuring Wildcard Tunnels....591
8.9.4 Encapsulation Types....592

Chapter 9: Integrating vCenter with DMF....593
9.1 Overview....594
9.2 Using SPAN to Monitor VMs....595
9.3 Using ERSPAN to Monitor VMs....596
9.3.1 Configuration Summary for vCenter Integration....597
9.4 Defining a Tunnel Endpoint....598
9.5 Integrate a vCenter Instance....599
9.5.1 Using a vCenter Instance as the Traffic Source in a DMF Policy....602
9.6 View vCenter Configuration....608
9.7 Integrating vCenter with DMF using Mirror Stack....611
9.7.1 vCenter Configuration....612
9.7.2 Configuring DMF....617
9.7.3 Limitations....618
9.7.4 Resources....619
9.8 Create Wildcard Tunnels for VMware vCenter Monitoring....620
9.8.1 Limitations....622
9.9 Minimum Permissions for Non-admin Users....623
9.10 Monitor vCenter Traffic by VM Names....629
9.10.1 Limitations....634
9.11 Using the Command Line Interface....635
9.11.1 Defining a Tunnel Endpoint....635
9.11.2 Integrate a vCenter Instance....636
9.11.3 View vCenter Configuration....642
9.11.4 Integrating vCenter with DMF using Mirror Stack....642
9.11.5 Create Wildcard Tunnels....645
9.11.6 Monitor vCenter Traffic by VM Names....647

Chapter 10: Integrating VMware NSX with DMF..... 650
10.1 Overview..... 651
10.2 NSX Compatibility and Requirements..... 652
10.3 Instance Configuration..... 653
10.3.1 VMware NSX Integration using the CLI..... 654
10.3.2 VMware NSX Integration using the GUI..... 656
10.4 Monitoring Configurations in Policies..... 660
10.4.1 Monitoring Configurations in Policies using the CLI..... 660
10.4.2 Monitoring Configurations in Policies using the GUI..... 662
10.5 Show Commands..... 664
10.6 Troubleshooting..... 667
10.7 Considerations..... 668
10.8 Resources..... 669

Chapter 11: CloudVision DMF Integration..... 670
11.1 Overview..... 671
11.2 Compatibility Requirements..... 672
11.3 CloudVision DMF Integration..... 673
11.4 Troubleshooting..... 679
11.5 Considerations..... 680
11.6 Resources..... 681
11.7 Using the Command Line Interface..... 682
11.7.1 CloudVision DMF Integration..... 683

Chapter 12: Advanced Fabric Settings..... 689
12.1 Configuring Advanced Fabric Settings..... 690
12.2 Managing VLAN Tags in the Monitoring Fabric..... 692
12.3 Auto VLAN Mode..... 694
12.3.1 Configuring Auto VLAN Mode using the CLI..... 696
12.3.2 Configuring Auto VLAN Mode using the GUI..... 698
12.4 Auto VLAN Range..... 701
12.5 Auto VLAN Strip..... 703
12.5.1 Auto VLAN Strip using the CLI..... 704
12.5.2 Auto VLAN Strip using the GUI..... 712
12.6 CRC Check..... 714
12.6.1 CRC Check using the CLI..... 715
12.6.2 CRC Check using the GUI..... 716
12.7 Custom Priority..... 718
12.7.1 Configuring Custom Priority using the GUI..... 719
12.7.2 Configuring Custom Priority using the CLI..... 721
12.8 Device Deployment Mode..... 722
12.8.1 Configuring Device Deployment Mode using the GUI..... 723
12.8.2 Configuring Device Deployment Mode using the CLI..... 725
12.9 Inport Mask..... 726
12.9.1 Inport Mask using the CLI..... 727
12.9.2 Inport Mask using the GUI..... 729
12.10 Match Mode..... 731
12.10.1 Setting the Match Mode Using the CLI..... 732
12.10.2 Setting the Match Mode Using the GUI..... 733
12.11 Retain User Policy VLAN..... 735
12.11.1 Retain User Policy VLAN using the CLI..... 736
12.11.2 Retain User Policy VLAN using the GUI..... 738

12.12 Tunneling..... 740
12.12.1 Configuring Tunneling using the GUI..... 741
12.12.2 Configuring Tunneling using the CLI..... 743
12.13 VLAN Preservation..... 744
12.13.1 Configure VLAN Preservation..... 745
12.13.2 Using the CLI to Configure VLAN Preservation..... 749
12.13.3 Troubleshooting..... 751
12.14 Reuse of Policy VLANs..... 753
12.15 Rewriting the VLAN ID for a Filter Interface..... 754
12.16 Reusing Filter Interface VLAN IDs..... 755
12.17 Using Push-per-filter Mode..... 757
12.18 Tag-based Forwarding..... 758
12.19 Policy Rule Optimization..... 759
12.19.1 Prefix Optimization..... 760
12.19.2 Transport Port Range and VLAN Range Optimization..... 761
12.20 Switch Dual Management Port..... 763
12.20.1 Overview..... 764
12.20.2 Configuring Dual Management Using the CLI..... 765
12.20.3 Configuring Dual Management Using the GUI..... 766
12.20.4 Management Interface Selection Using the GUI..... 770
12.20.5 Management Interface Selection Using the CLI..... 773
12.20.6 Switch Fabric Management Redundancy Status..... 774
12.20.7 Additional Notes..... 775
12.20.8 Management Redundancy on EOS Fixed System Chassis..... 776
12.21 Controller Lockdown..... 786
12.22 CPU Queue Stats and Debug Counters..... 787
12.22.1 Troubleshooting..... 790
12.22.2 Limitations..... 791
12.23 Egress Filtering..... 792
12.23.1 Configuring Egress Filtering using the CLI..... 793
12.23.2 Configuring Egress Filtering using the GUI..... 797
12.23.3 Syslog Messages..... 802
12.23.4 Troubleshooting..... 803
12.23.5 Limitations..... 804

Chapter 13: Advanced Policy Configuration..... 805

13.1 Advanced Match Rules..... 806
13.1.1 Match Fields and Criteria..... 807
13.1.2 Match-except Rules..... 809
13.1.3 Matching with IPv6 Addresses..... 810
13.1.4 Port and VLAN Range Matches..... 811
13.1.5 User Defined Filters..... 813
13.2 Using the Filter and Delivery Role with MAC Loopback for a Two-stage Policy..... 815
13.2.1 Configure Filter and Delivery Interfaces with MAC Loopback..... 817
13.2.2 Using the CLI To Configure a Filter and Delivery Interface with MAC Loopback..... 820
13.3 Rate Limiting Traffic to Delivery Interfaces..... 821
13.3.1 Rate Limiting Using the GUI..... 822
13.3.2 Rate Limiting Using the CLI..... 824
13.4 Configuring Overlapping Policies..... 825
13.4.1 Configuring the Policy Overlap Limit Using the GUI..... 827
13.4.2 Configuring the Overlapping Policy Limit Using the CLI..... 829
13.4.3 Using the CLI to View Overlapping Policies..... 830
13.4.4 Configuring the Policy Overlap Limit Strict using the GUI..... 831
13.4.5 Configuring the Policy Overlap Limit Strict using the CLI..... 833
13.4.6 Exclude Inactive Policies from Overlap Limit Calculation..... 834

13.4.7 Exclude Expired Policies from Overlap Limit Calculation....836
13.5 Viewing Information about Policies....839
13.5.1 Viewing Policy Flows....840
13.5.2 Viewing Packets Dropped by Policies....841
13.6 Using Rule Groups....842
13.6.1 Using the GUI to Configure Rule Groups....843
13.6.2 Using the CLI to Configure Interface Groups....845
13.7 PTP Timestamping....846
13.7.1 Platform Compatibility....847
13.7.2 Configuration....848
13.7.3 Configuring PTP Timestamping using the CLI....849
13.7.4 Configuring PTP Timestamping using the GUI....850
13.7.5 PTP Interface Configuration....856
13.7.6 Policy Configuration for Timestamping....857
13.7.7 L2GRE Encapsulation of Packets with Arista Timestamp Headers....858
13.7.8 Using the CLI Show Commands....859
13.7.9 Configuration Validation Messages....862
13.7.10 Troubleshooting....863
13.7.11 Limitations....864

Appendix A: Stenographer Reference for DMF Recorder Node....865
A.1 Stenographer Query Syntax....866
A.2 Example Stenographer Queries....869

Appendix B: DMF Recorder Node REST API....871
B.1 Authentication....872
B.1.1 Basic HTTP Authentication....873
B.1.2 Authentication with an Authentication Token....874
B.2 DMF Recorder Node API Headers....876
B.3 DMF Recorder Node REST APIs....877
B.3.1 Ready....878
B.3.2 Query Window....879
B.3.3 Query Size....880
B.3.4 Query Application....881
B.3.5 Query Packet....882
B.3.6 Query Analysis Filter Stenographer Analysis Type....883
B.3.7 Query Replay Request Stenographer Real-time Boolean....884
B.3.8 Query Replay Request Filter Stenographer Integer....885
B.3.9 Query Replay Done....886
B.3.10 Erase Packet Filter Stenographer Query String....887
B.3.11 Event Update Trigger Boolean Name Integer....888
B.3.12 Abort Query....889
B.3.13 Abort All Query....890
B.3.14 Queries....891
B.3.15 Status Query....892
B.3.16 Status All....893
B.4 Recorder Node Recording State API....894

Appendix C: DMF Controller in Microsoft Azure....897
C.1 Configuration....898
C.2 Syslog Messages....901
C.3 Troubleshooting....902
C.4 Limitations....903

C.5 Resources..... 904
C.6 Diagrams..... 905

Appendix D: DMF Controllers in Google Cloud VMware Engine.....906
D.1 Firstboot Parameters.....913
D.2 Post Installation Verification.....915
D.3 Troubleshooting.....916
D.4 Limitations..... 917

Appendix E: Telemetry Collector.....918
E.1 Storage Health for CVA 7.1.0.....922
E.1.1 Troubleshooting..... 925
E.1.2 Considerations.....925

Appendix F: DMF GUI REST API Inspector.....926

Appendix G: Configuring Third-Party Services..... 935
G.1 Services in the DANZ Monitoring Fabric.....936
G.1.1 Configure a DMF Unmanaged Service.....938
G.1.2 Configure a DMF Unmanaged Service using the CLI..... 941
G.2 Service Insertion and Chaining in a DMF Policy.....941

Appendix H: References.....943
H.1 Related Documents.....944

Chapter 1

Introduction to DMF and Overview

This chapter introduces the DANZ Monitoring Fabric (DMF) and the user interfaces for out-of-band monitoring and configuration.

DANZ Monitoring Fabric (DMF) is a cloud-first Network Packet Broker (NPB) that provides a single pane of glass with an integrated visibility fabric. The DMF solution includes NPB functionality with the DMF Recorder Node and the Analytics Node for deeper monitoring and pervasive security of out-of-band workloads in hybrid cloud deployments.

DMF leverages an SDN-controlled fabric using high-performance, open networking (white box/brite box) switches and industry-standard x86 servers to deploy highly scalable and flexible network visibility and security solutions. Traditional, box-based, hardware-centric NPBs are architecturally limited when trying to meet the evolving security and visibility demands of Cloud Native data centers. DMF addresses the challenges of traditional NPB solutions by enabling a scale-out fabric for enterprise-wide security and monitoring, a single pane of glass for operational simplicity, and multi-tenancy for multiple IT teams, including NetOps, DevOps, and SecOps.

1.1 Out-of-band Monitoring with DANZ Monitoring Fabric

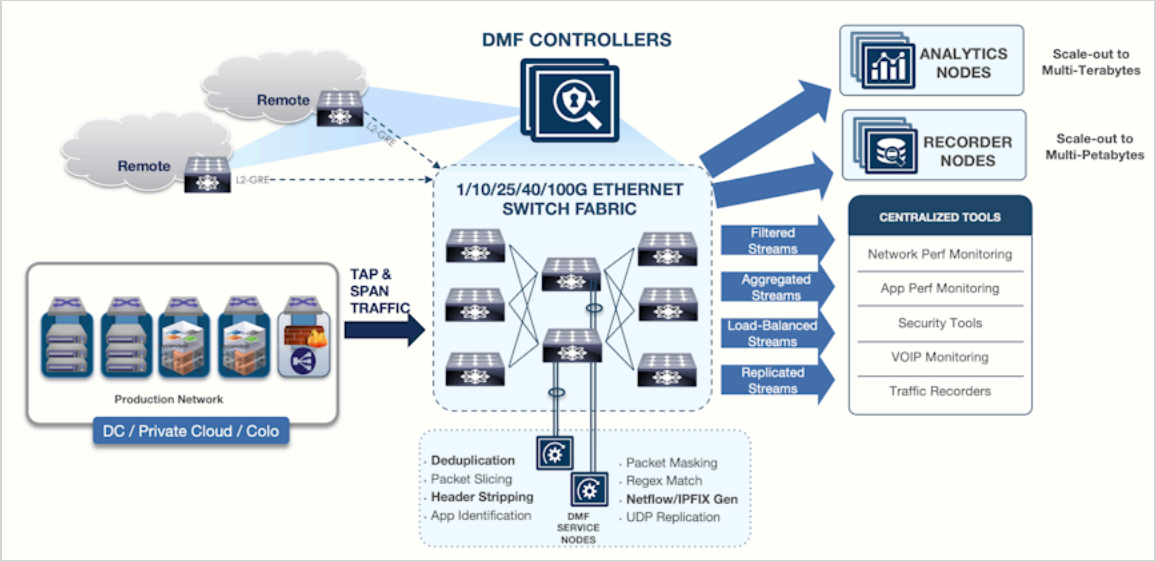
As data center networks move toward 40/100G designs, cloud computing, hyper scale data analytics, and 5G mobile services, traffic monitoring must transition to next-generation designs. To manage the modern data center, much network traffic must be copied and aggregated from TAP or SPAN ports and forwarded to monitoring and analysis tools. These tools, used in managing network performance, application performance, security, and compliance, leverage other systems such as data recorders, intrusion detection systems, data leakage detectors, SLA measurement devices, and other traffic analyzers like Wireshark.

DANZ Monitoring Fabric (DMF) uses high-performance open-networking switches to deliver an open, production-grade, and scalable monitoring solution based on Software Defined Networking (SDN) technology. The centralized DMF Controller provides flexibility, simplifies policy management and monitoring fabric configuration, and supports cost-effective monitoring of data centers and remote sites or branches with up to several thousand TAP and SPAN ports.

DMF architecture, inspired by hyper scale networking designs, consists of the following components:

- HA pair of SDN-enabled DMF Controllers (VMs or hardware appliances), enable simplified and centralized configuration, monitoring, and troubleshooting.
- Arista Networks SDN-enabled Switch Light OS is a production-grade, lightweight OS that runs on DMF Ethernet switches.
- DANZ Service Nodes (optional), a Data Plane Development Kit (DPDK)-powered, x86-based appliances that connect to the DMF singly or as part of a service node chain. The service node provides advanced packet functions, such as deduplication, packet slicing, header stripping, regex matching, packet masking, UDP replication, and IPFIX/NetFlow generation.
- DANZ Recorder Nodes (optional) are x86-based appliances connected to the DMF and are managed via the DMF Controller to provide petabyte packet recording, querying, and replay functions.
- Analytics Nodes (optional) are x86-based appliances that integrate with the DMF to provide multi-terabit, security, and performance analytics with configurable, historical time-series dashboards.

Figure 1-1: Out-of-Band Monitoring with DANZ Monitoring Fabric



DMF lets a network operator easily deploy data center-wide monitoring with the following benefits:

- Organization-wide visibility: delivers traffic from any TAP to any tool at any time across one or multiple locations.
- Flexible, scale-out fabric deployment: supports a large number of 1G, 10G, 25G, 40G, and 100G ports (thousands per fabric).
- Multi-tenant tap and tool sharing: supports monitoring by multiple teams to enable Monitoring Fabric as a Service.
- Massive operational simplification: provides a single pane of glass for provisioning, management, monitoring, and debugging through a centralized SDN Controller. This feature eliminates needing a box-by-box configuration.
- Centralized programmability: a REST-based API architecture enables event-based, centralized policy management and automation for integrated end-to-end IT work flows. This feature leverages DMF Service Nodes, Analytics Nodes, and Recorder Nodes.
- Dramatic cost savings: Achieving a significant reduction in the total cost of ownership by using open Ethernet switches in combination with industry-standard x86 servers, optimized usage of tools, and SDN-enabled operations and automation.

1.2 Using the DANZ Monitoring Fabric GUI

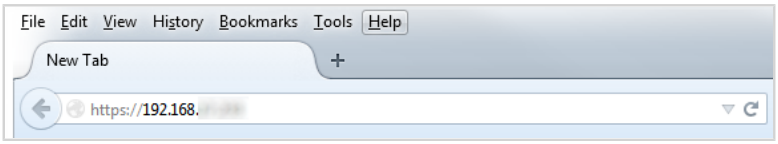
The DANZ Monitoring Fabric (DMF) Graphical User Interface (GUI) performs similar operations to the CLI using a graphic user interface instead of text commands and options. The DMF GUI is compatible with recent versions of any of the following supported browsers:

- Mozilla Firefox
- Google Chrome
- Microsoft Edge
- Internet Explorer
- Apple Safari

To connect to the DMF GUI, use the DMF Controller IP address. Use the virtual IP (VIP) assigned to the cluster if configured during deployment. Using the VIP ensures that the user connects to the current active Controller, regardless of any failover that may have occurred.

Use the active Controller for all configuration operations and to obtain reliable information when monitoring DMF. The standby Controller is provided only for redundancy if the active Controller becomes unavailable. Do not perform any configuration using the standby Controller, and any information displayed may not be accurate. The figure below illustrates connecting to the DMF GUI using HTTPS (***port 443***) at the IP address ***192.168.17.233***

Figure 1-2: Connecting to the DANZ Monitoring Fabric GUI

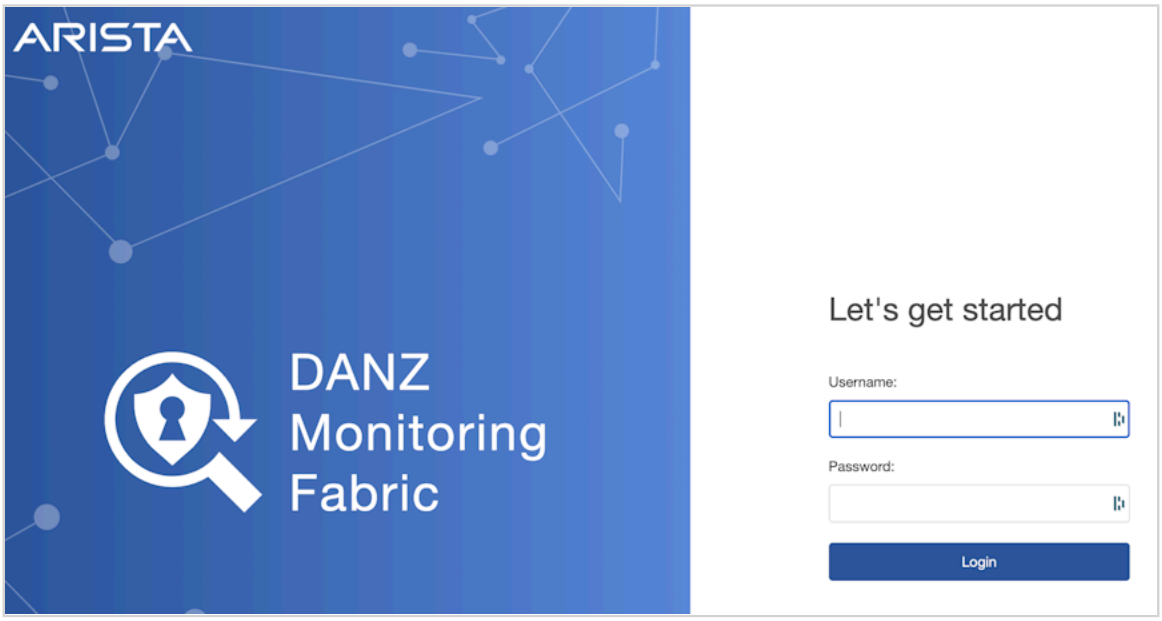


Connecting to the Controller for the first time may result in a security exception prompt (message) because the Controller HTTPS server uses an unknown (self-signed) certificate authority.

Note: When using Internet Explorer, the login attempt may fail if the system time is different than the Controller time. To remedy this, ensure the system used to log in to the Controller is synchronized with the Controller.

After accepting the prompts, the system displays the login prompt, shown in the figure below.

Figure 1-3: DANZ Monitoring Fabric GUI Login Prompt



Use the admin username and password configured for the DMF Controller during installation or any user account and password configured with administrator privileges. A user in the read-only group will have access to options for monitoring fabric configuration and activity but cannot change the configuration.

Figure 1-4: DANZ Monitoring Fabric GUI Main Menu



When logging in to the DMF GUI, a landing page appears. This page shows the **DMF Controller Overview**, dashboard, and a menu bar (pictured above) with sub-menus containing options for setting up DMF and monitoring network activity. The menu bar includes the following sub-menus:

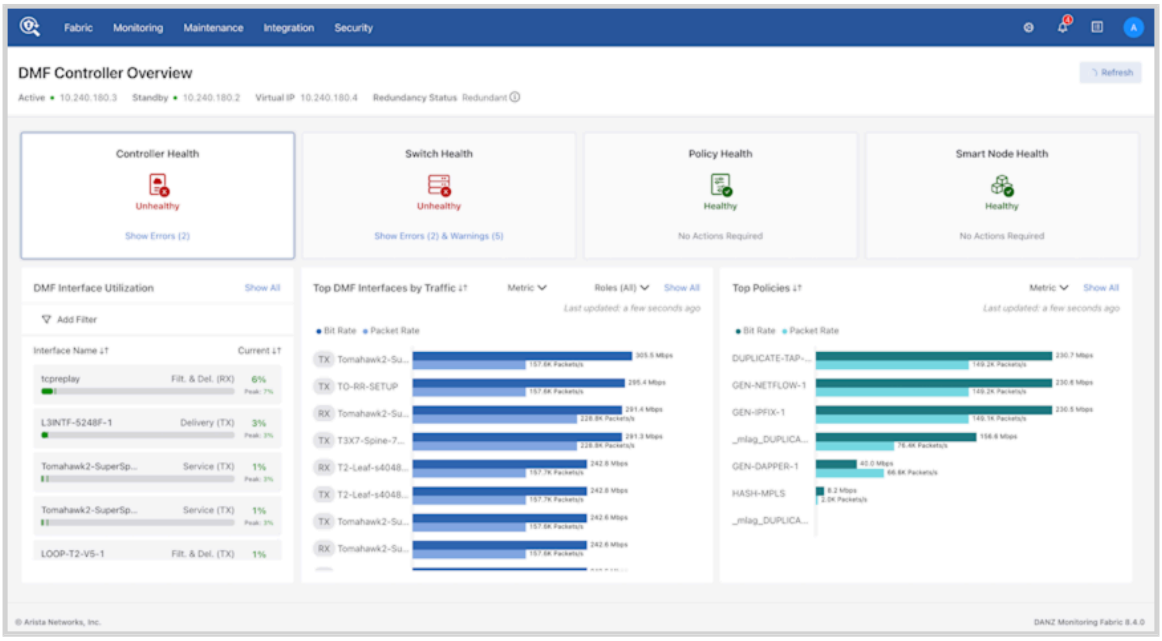
- **Fabric**: Manage DMF switches and interfaces.
- **Monitoring**: Manage DMF policies, services, and interfaces.
- **Maintenance**: Configure fabric-wide settings (clock, SNMP, AAA, sFlow®*, Logging, Analytics Configuration).
- **Integration**: Manage the integration of vCenter instances to allow monitoring traffic using DMF.
- **Security**: Manage administrative access.
- **DMF Features**: Select the **gear** icon in the navigation bar to view the DMF Features page.
- **Alerts**: Select the **Notification Bell** to view the Alerts page.
- **Background Tasks**: Select the **Task** icon to view running background tasks.

* sFlow® is a registered trademark of Inmon Corp.

- **Admin** (**A**): A page that displays or edits user preferences; the ability to change the password or sign out. Set **Display Preferences**, enter the **API Inspector**, change **Settings**, and view **About** details.

The newly designed dashboard displays information about the Controller, including switches, interfaces, policies, and Smart Nodes.

Figure 1-5: DMF Controller Overview



The header displays the following basic information about the Controller:

- Active IP address
- Standby IP address
- Virtual IP address
- Redundancy Status - The status contains an informational tool tip that can be hovered for more details.

Four cards control the type of content displayed on the main section of the page. The cards are:

- Controller Health
- Switch Health
- Policy Health
- Smart Node Health

1.2.1 Admin Settings

The Admin settings menu provides access to the following features and functions.

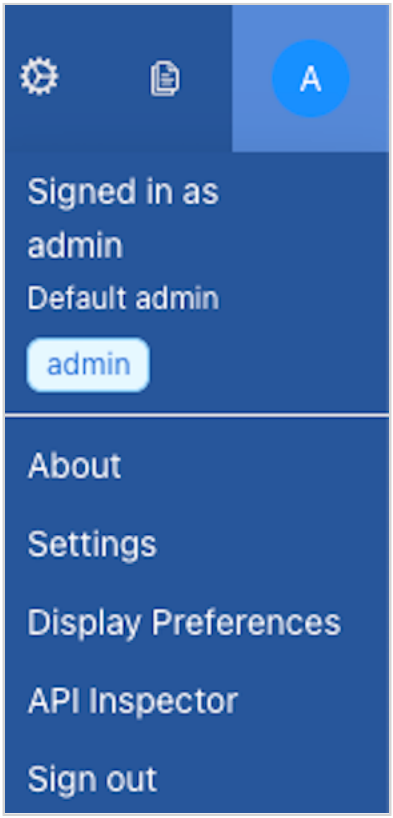
- **Account Sign-in and User Role**
- **About**
- **Settings**
- **Display Preferences**
- **API Inspector**
- **Sign out**

Hover over and select the A (user) icon to view the drop-down menu.

Figure 1-6: Gear Icon

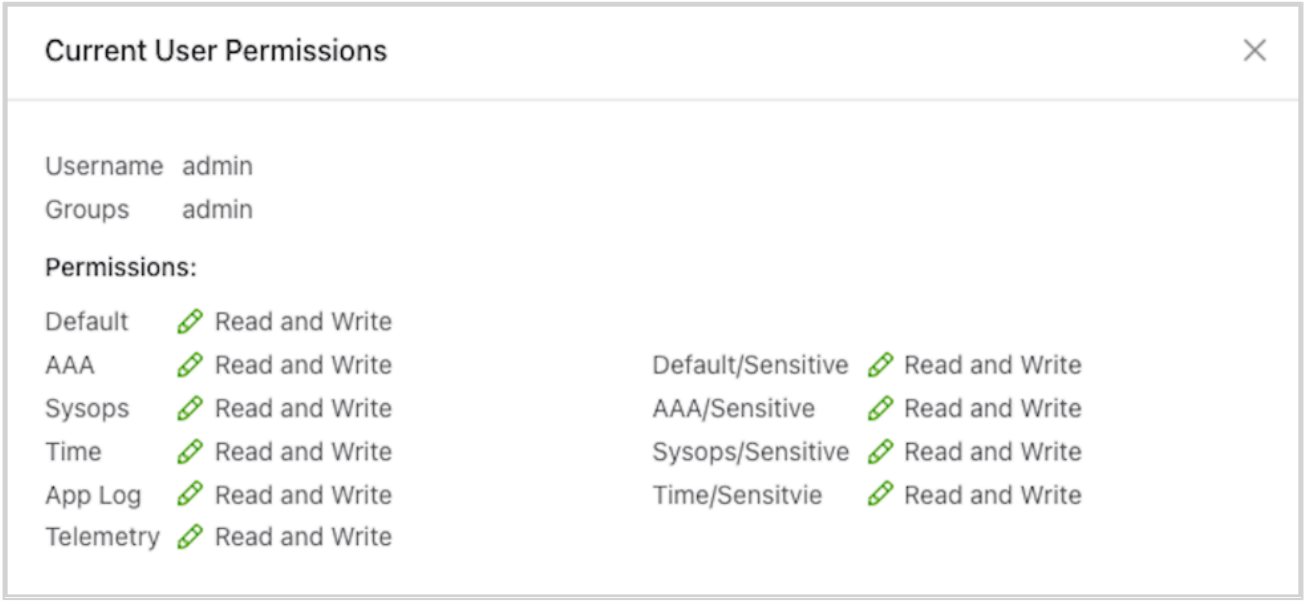


Figure 1-7: Admin Menu



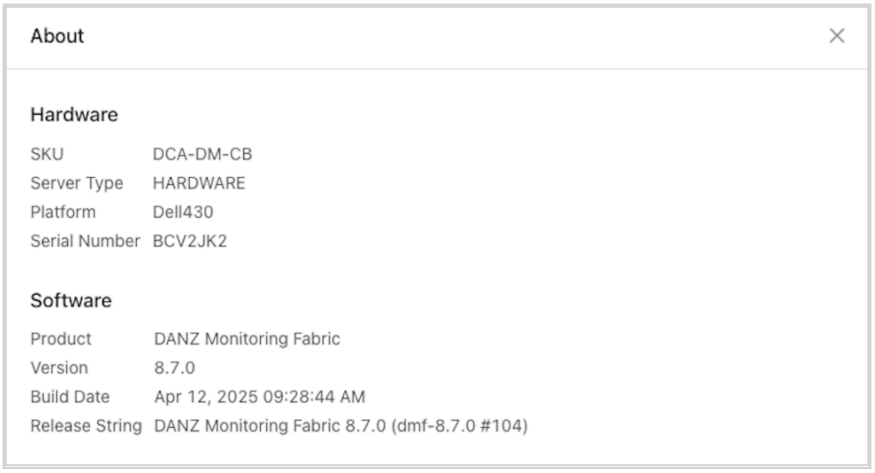
Selecting the **User Account** icon opens a **Current Users Permissions** window.

Figure 1-8: Current User Permissions



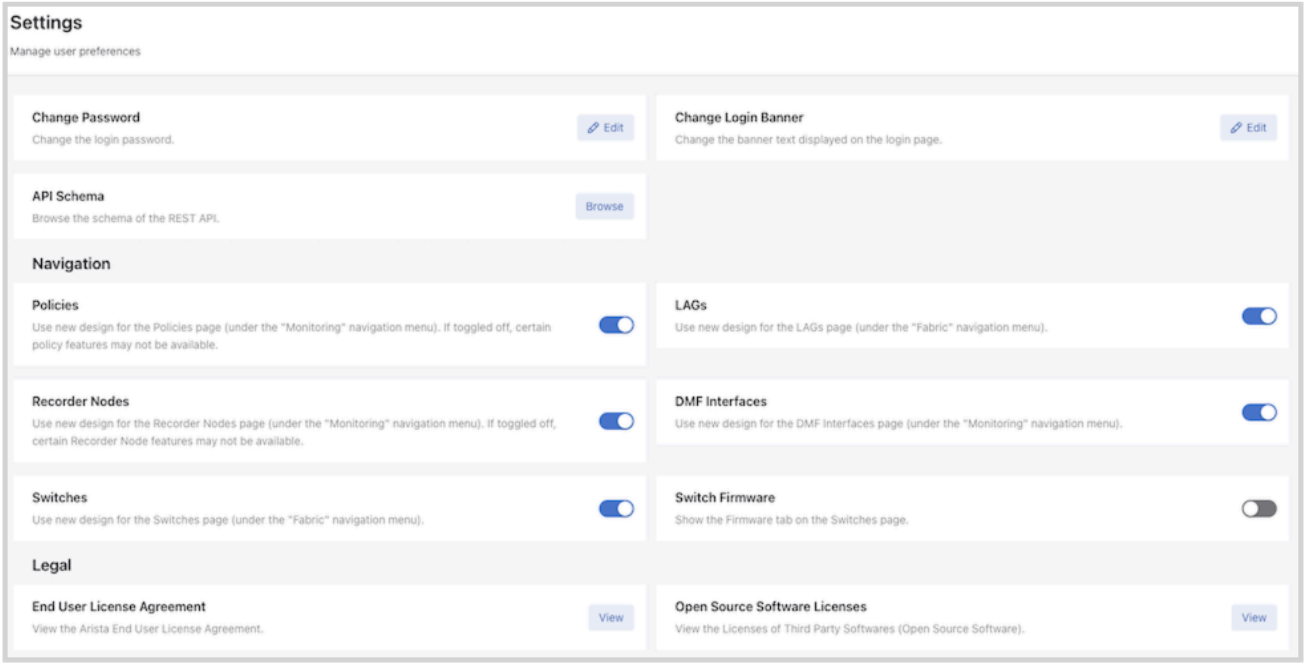
About: Open an information window providing details about the DMF Instance.

Figure 1-9: Hardware / Software Information



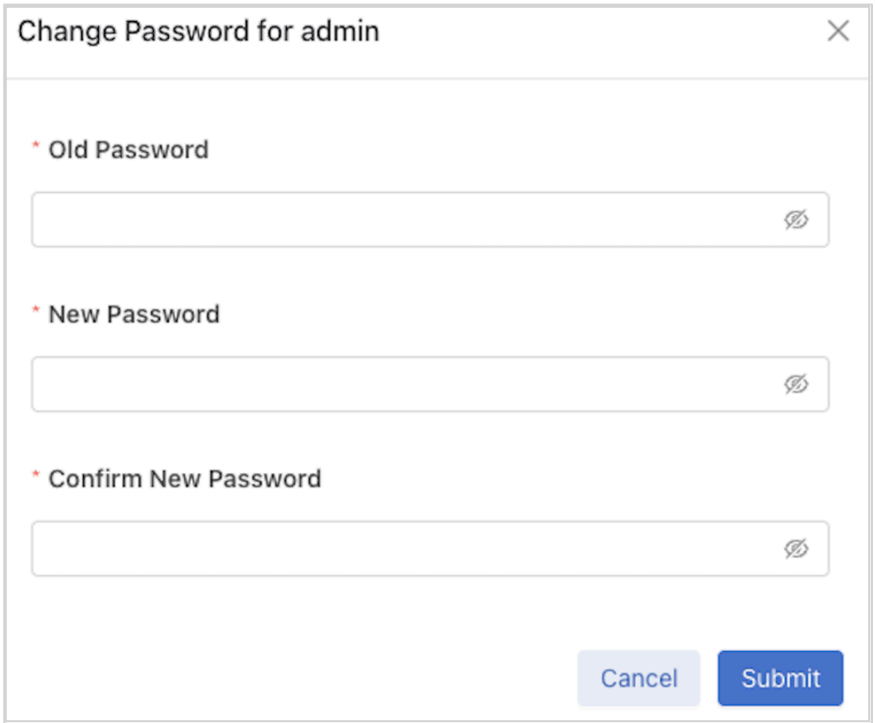
To manage user preferences select **Settings**.

Figure 1-10: Settings



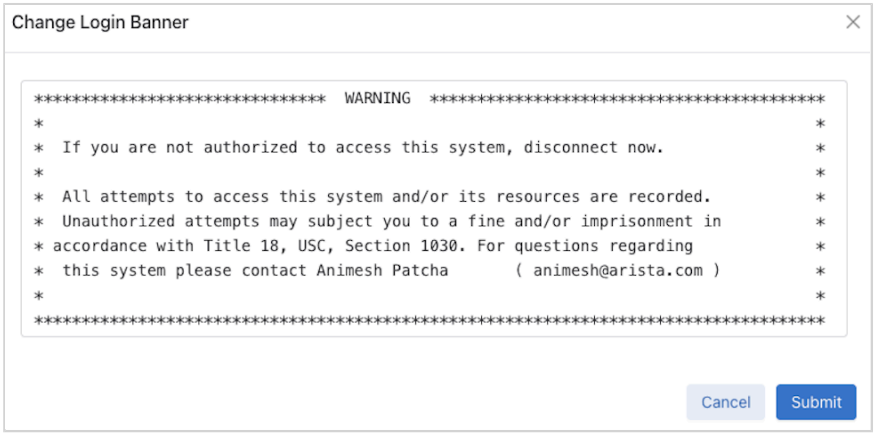
- **Change Password**: Change the existing user account password.

  Figure 1-11: Password Change



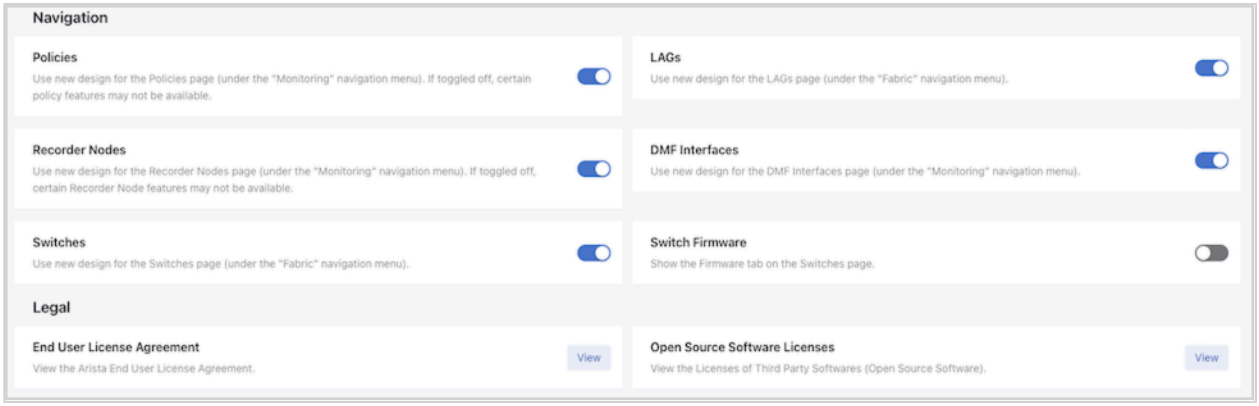
- **Change Login Banner**: Change the login banner that appears when accessing DMF.

Figure 1-12: Change Banner



- **Navigation and Legal**: Configure the user interface preferences and view license details.

Figure 1-13: Navigation



Display Preferences: Review the preferred **Bit Rate Units** for data display.

Figure 1-14: Display Preferences



API Schema: Browse the API Schema of the DMF API.

Figure 1-15: API Schema



Sign out: Logout of the DMF Instance.

1.2.2 DMF Features Page

To navigate the **DMF Features Page**, select the **gear** icon in the navigation bar.

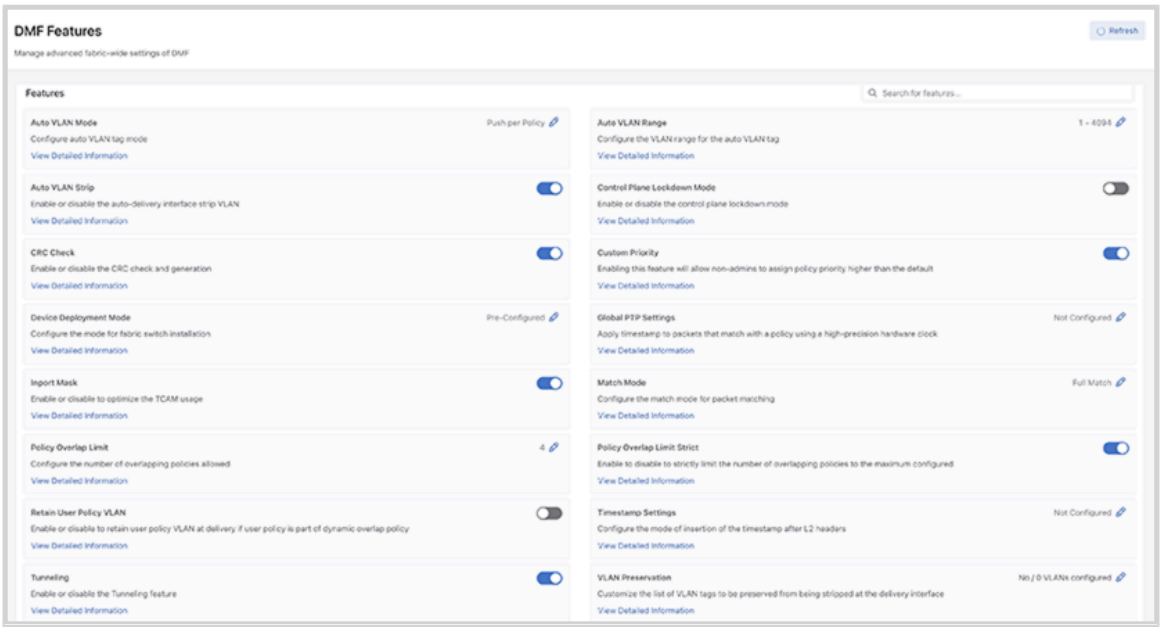
Figure 1-16: Gear Icon



Page Layout

All fabric-wide configuration settings required in advanced use cases for deploying DMF policies appear in the new DMF Features Page.

Figure 1-17: DMF Features Page



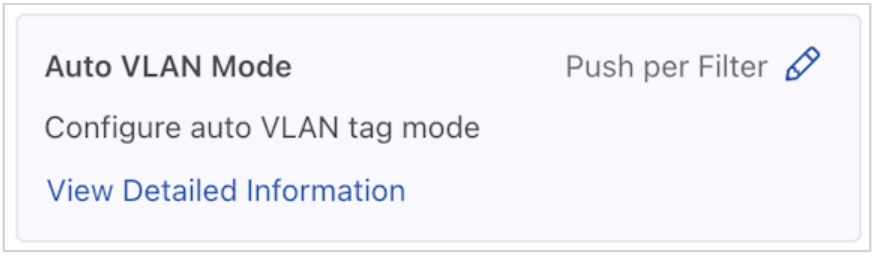
The fabric-wide options used with DMF policies include the following:

Table 1: Feature Set

Auto VLAN Mode	Auto VLAN Range
Auto VLAN Strip	Control Plane Lockdown Mode
CRC Check	Custom Priority
Device Deployment Mode	Error Disable
Filter MAC Rewrite	Global PTP Settings
Inport Mask	MACsec
Match Mode	Policy Overlap Limit
Policy Overlap Limit Strict	Retain User Policy VLAN
Supervisor Redundancy Settings	Timestamp Settings
Tunneling	VLAN Preservation

Each card on the page corresponds to a feature set.

Figure 1-18: Feature Set Card



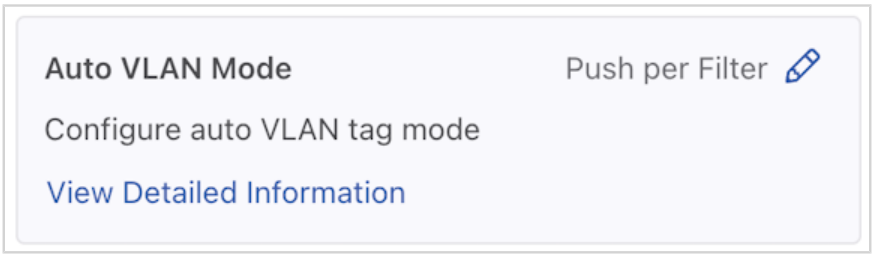
The UI displays the following:

- Feature Title
- A brief description
- View / Hide Detailed Information
- Current Setting
- Edit Link - Use **Edit** (**pencil** icon) to change the value.

View Detailed Information

Each configuration option has detailed information. For more details, select the **View Detailed Information** link on each card.

Figure 1-19: View Detailed Information

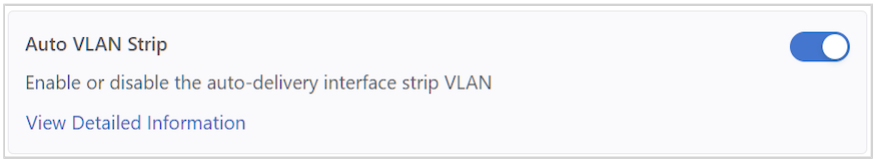


Feature Settings

Auto VLAN Strip

1. A **toggle button** controls the configuration of this feature. Locate the corresponding card and move the **toggle** button.

   Figure 1-20: Toggle Switch



2. A confirm window pops up, displaying the corresponding prompt message. Select **Enable** to confirm the configuration changes or choose **Cancel** to cancel the configuration. Conversely, to disable the configuration, select **Disable**.

Figure 1-21: Confirm Enable

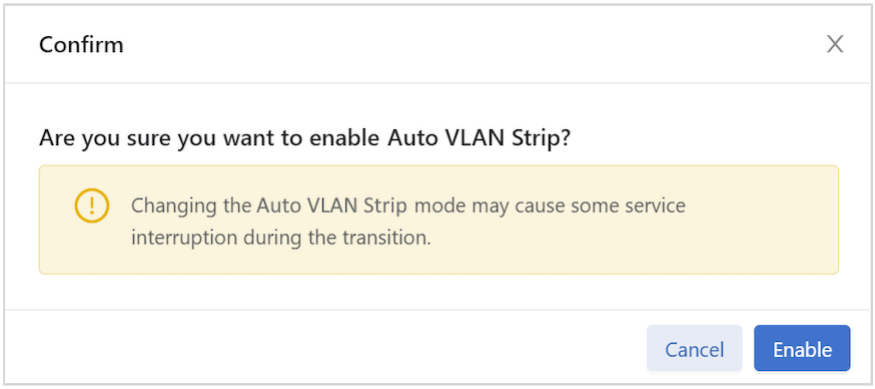
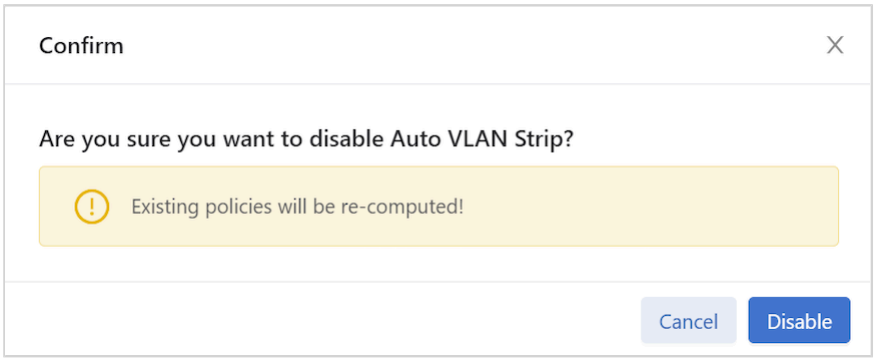
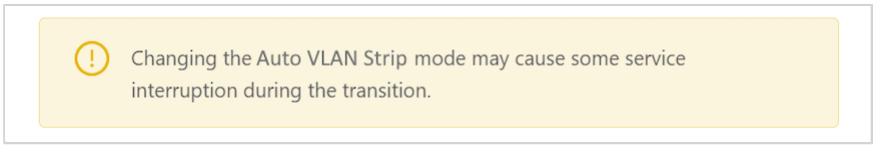


Figure 1-22: Confirm Disable



3. Review any warning messages that appear in the confirmation window during the configuration process.

Figure 1-23: Warning Message - Changing



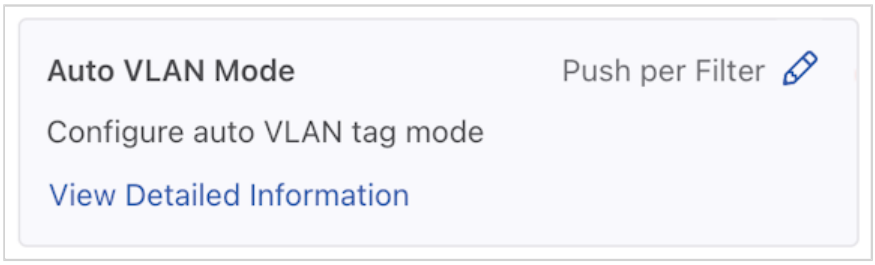
The following feature sets work in the same manner as the **Auto VLAN Strip** feature described above.

CRC Check	Policy Overlap Limit Strict
Custom Priority	Retain User Policy VLAN
Inport Mask	Tunneling

Auto VLAN Mode

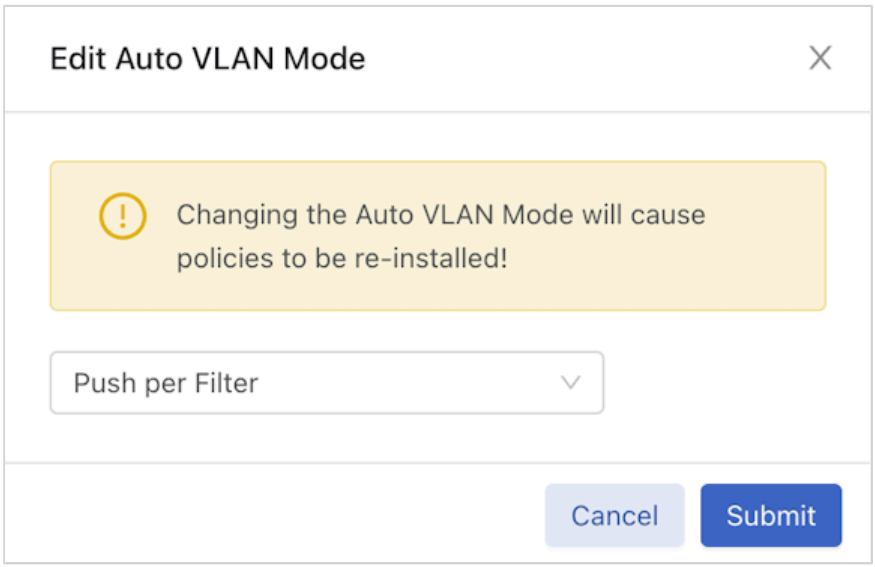
1. Control the configuration of this feature using the **Edit** icon by locating the corresponding card and selecting the **pencil** icon.

   Figure 1-24: Auto VLAN Mode Config



2. A confirmation edit dialogue window appears, displaying the corresponding prompt message.

   Figure 1-25: Edit VLAN Mode



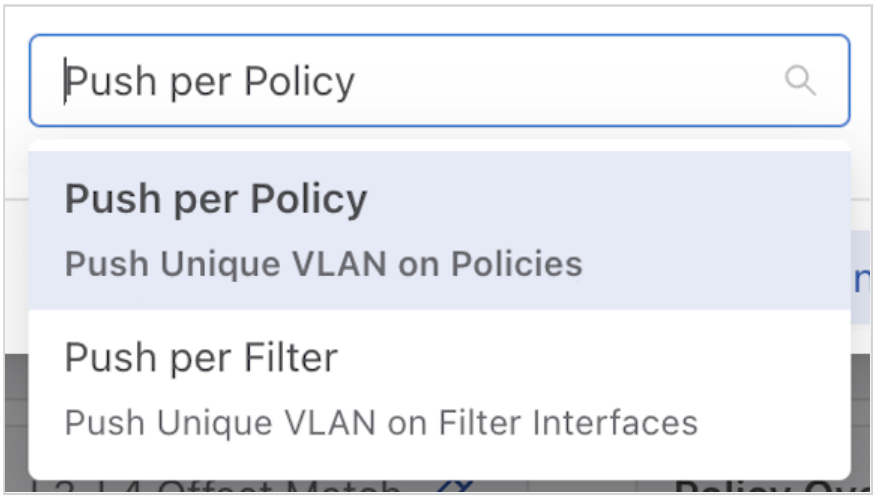
3. To configure different modes, select the drop-down arrow to open the menu.

   Figure 1-26: Drop-down Example



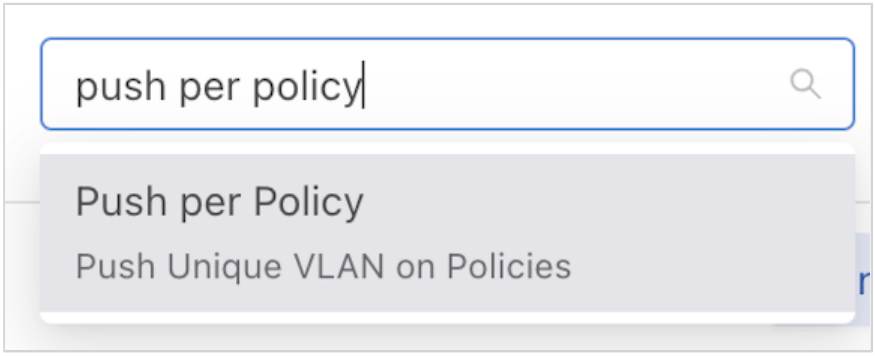
4. From the drop-down menu, select and select the desired mode.

Figure 1-27: Push Per Policy



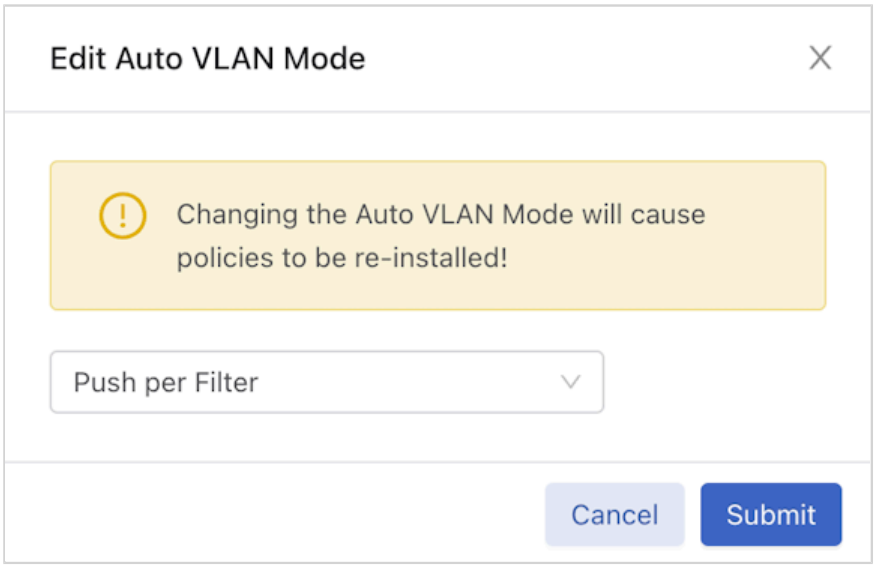
5. Alternatively, enter the desired mode name in the input area.

Figure 1-28: Push Per Policy



6. Select **Submit** to confirm the configuration changes or choose **Cancel** to discard the changes.

Figure 1-29: Submit Button



7. After successfully setting the configuration, the current configuration status displays next to the **edit** icon.

Figure 1-30: Current Configuration Status



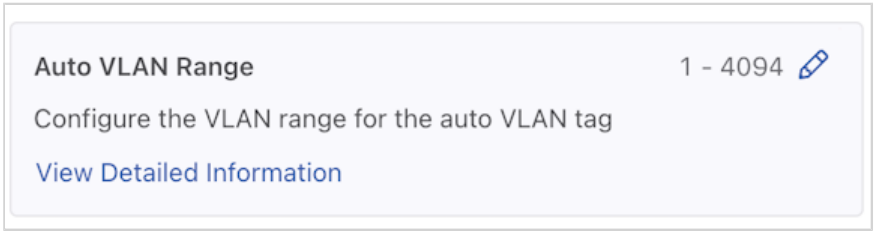
The following feature sets work in the same manner as the **Auto VLAN Mode** feature described above.

- **Device Deployment Mode**
- **Match Mode**

Auto VLAN Range

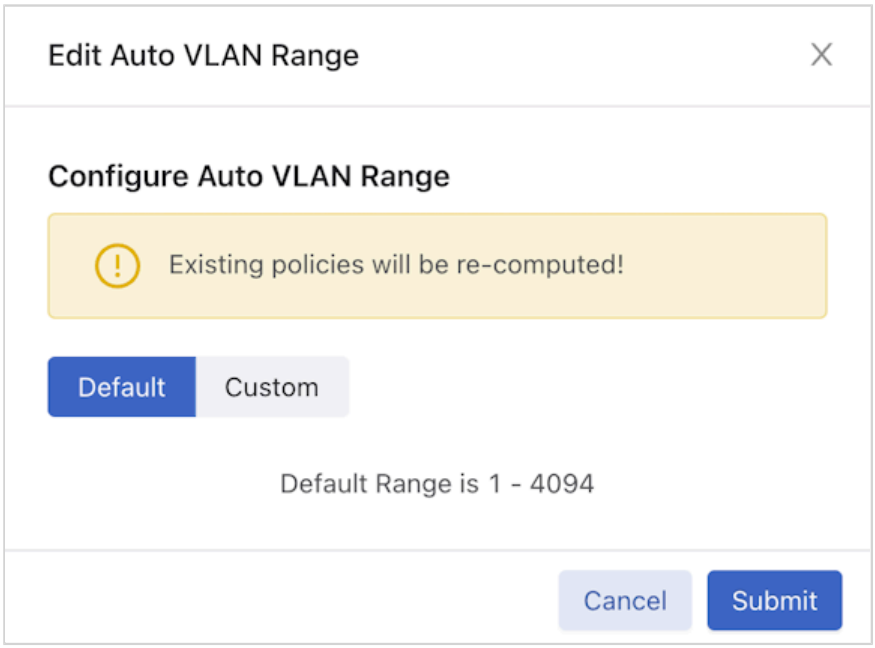
1. Control the configuration of this feature using the **Edit** icon by locating the corresponding card and selecting the **pencil** icon.

Figure 1-31: Edit Auto VLAN Range



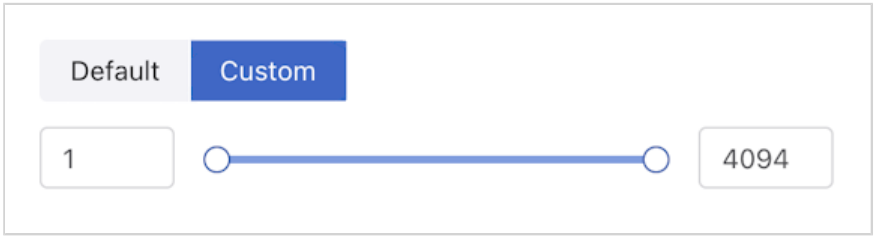
2. A configuration edit dialogue window pops up, displaying the corresponding prompt message. The Auto VLAN Range defaults to 1 - 4094.

   Figure 1-32: Edit Auto VLAN Range



3. Select **Custom** to configure the custom range.

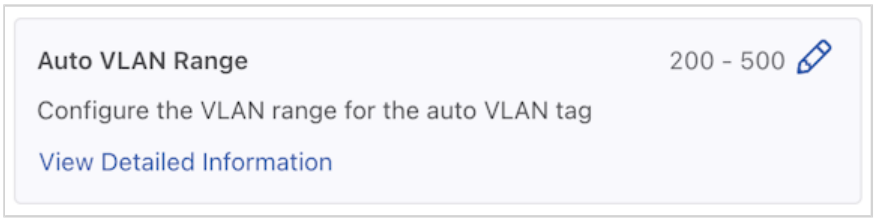
   Figure 1-33: Custom Button



4. Adjust range value (minimum value: 1, maximum value: 4094). There are three ways to adjust the value of a range:
   - Directly enter the desired value in the input area, with the left side representing the minimum value of the range and the right side representing the maximum value.
   - Adjust the value by dragging the **slider** using a mouse. The left knob represents the minimum value of the range, while the right knob represents the maximum value.
   - Use the up and down arrow buttons in the input area to adjust the value accordingly. Pressing the up arrow increments the value by 1, while pressing the down arrow decrements it by 1.
5. Select **Submit** to confirm the configuration changes or **Cancel** to discard the changes.

6. After successfully setting the configuration, the current configuration status displays next to the **pencil** (edit) icon.

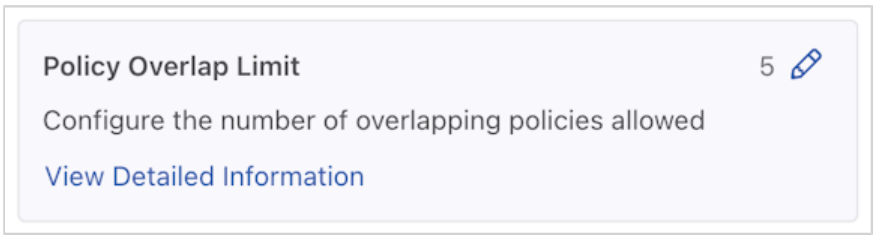
Figure 1-34: Configuration Change Success



Policy Overlap Limit

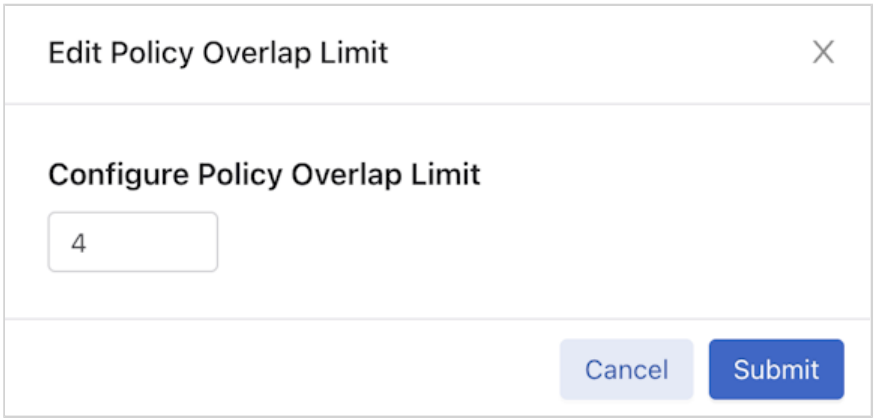
1. Control the configuration of this feature using the **Edit** icon by locating the corresponding card and selecting the **pencil** icon.

Figure 1-35: Policy Overlap Limit



2. A configuration edit dialogue window pops up, displaying the corresponding prompt message. By default, the Policy Overlap Limit is 4.

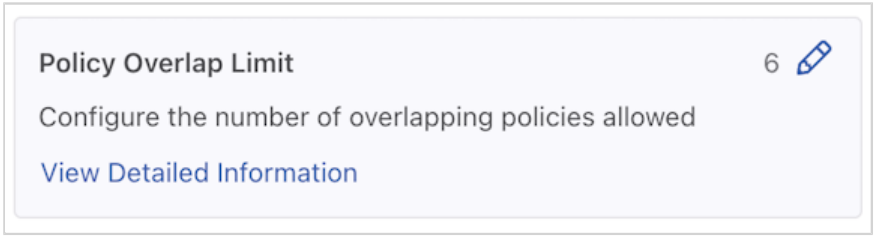
Figure 1-36: Edit Policy Overlap Limit



3. Adjust the Value (minimum value: 0, maximum value: 10). There are two ways to adjust the value:
   - Directly enter the desired value in the input area.
   - Use the up and down arrow buttons in the input area to adjust the value accordingly. Pressing the up arrow increments the value by 1, while pressing the down arrow decrements it by 1.
4. Select **Submit** to confirm the configuration changes or **Cancel** to discard the changes.

5. After successfully setting the configuration, the current configuration status displays next to the **pencil** (edit) icon.

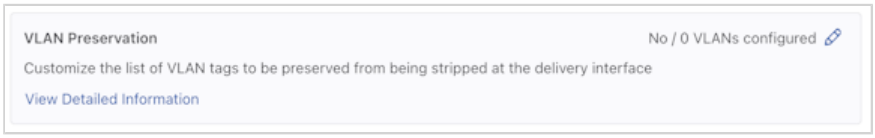
Figure 1-37: Policy Overlap Limit Change Success



VLAN Preservation

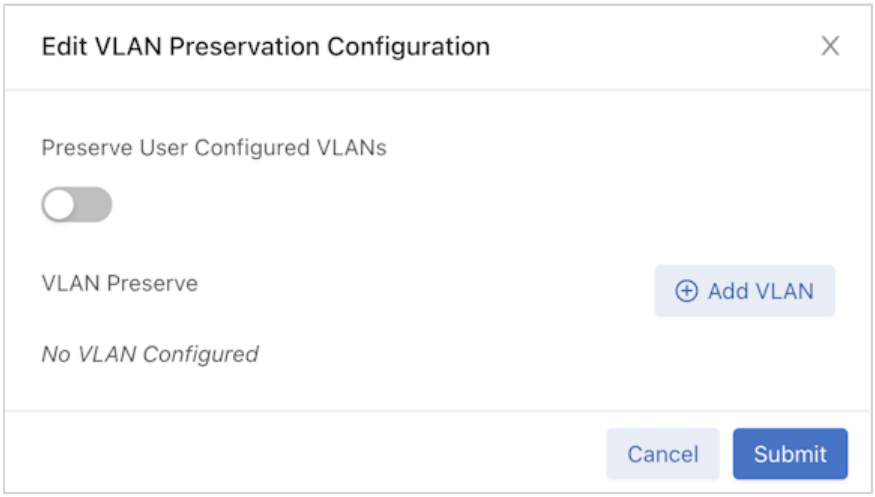
1. Control the configuration of this feature using the **Edit** icon by locating the corresponding card and selecting the **pencil** icon.

Figure 1-38: VLAN Preservation Feature Set



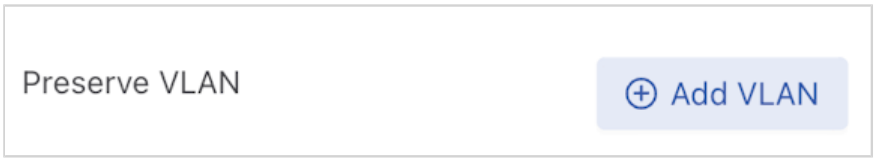
2. A configuration edit dialogue window appears displaying the corresponding prompt message. The VLAN Preservation defaults to:
   - Preserve User Configured VLANS: Off
   - Preserve VLAN: No VLAN Configured
3. To configure **Preserve User Configured VLANs**, **toggle** on the switch.

Figure 1-39: Edit VLAN Preservation Configuration



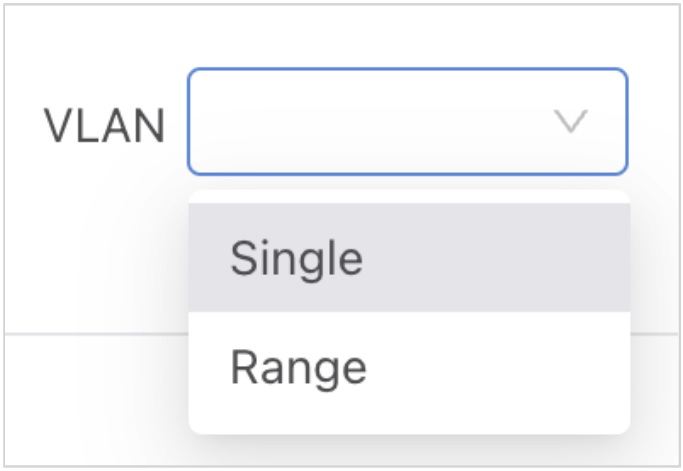
4. To configure **Preserve VLAN**, select **Add VLAN** to add a configuration area for preserving the VLAN value.

   Figure 1-40: Preserve VLAN - Add VLAN



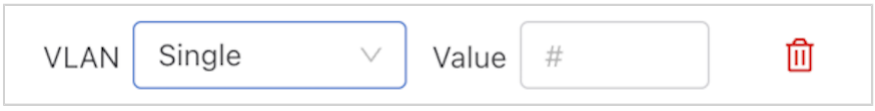
5. Select the drop-down icon. There are two ways to configure the preserved VLAN value (minimum value: 1, maximum value: 4094) and a method to delete an entry.

   Figure 1-41: VLAN Single Example



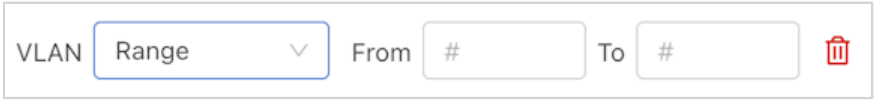
- **Add Single**: Choose **Single** in the VLAN drop-down menu, and type in the value in the input area.

  Figure 1-42: Add Single VLAN



- **Add Range**: Choose **Range** in the VLAN drop-down menu, and type in the input area's minimum and maximum values.

  Figure 1-43: Add VLAN Range

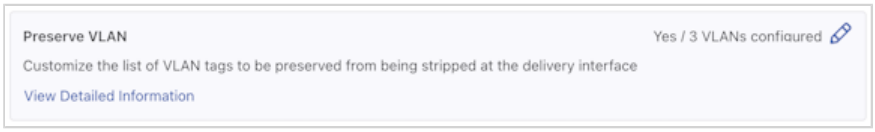


- **Delete**: Since there must be a corresponding number in the value input area when submitting the configuration, when accidentally adding multiple redundant VLAN configuration areas, delete the corresponding rows by selecting the red trash can icon .

> **Note:** The feature supports combinations of any number of single values and any number of range values.

6. Select **Submit** confirm the configuration changes or **Cancel** to discard the changes.
7. After successfully setting the configuration, the current configuration status displays next to the **pencil** (edit) icon.

Figure 1-44: Preserve VLAN Configuration Change



Global PTP Settings

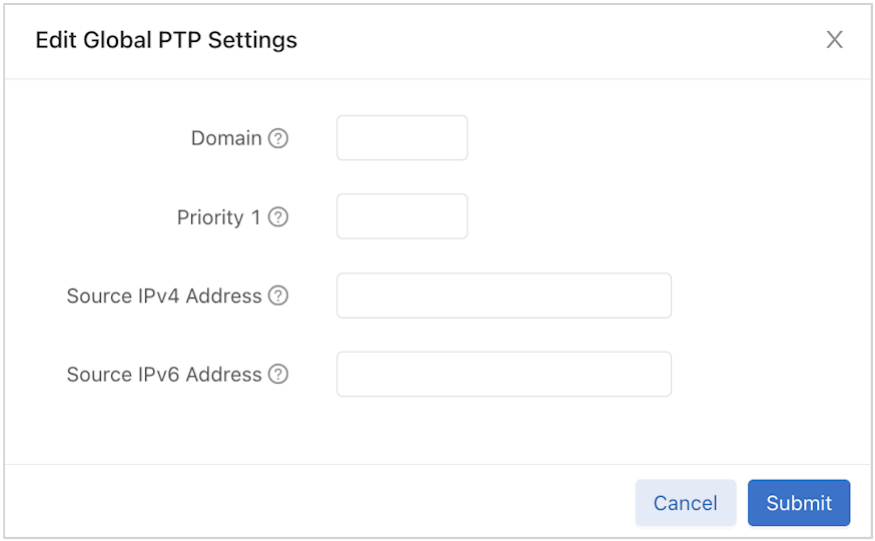
1. Control the configuration of this feature using the **Edit** icon by locating the corresponding card and selecting the **pencil** icon.

Figure 1-45: Global PTP Settings



2. A configuration edit dialogue window appears displaying the corresponding prompt message. By default, these features are not configured. Enter the desired configuration value in the corresponding input area. Hover over the **question mark** icon to obtain additional explanatory information.

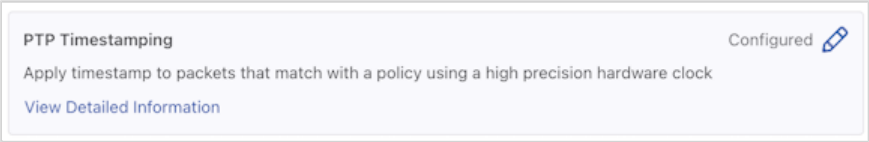
Figure 1-46: Edit PTP Settings



3. Select **Submit** to confirm the configuration changes or **Cancel** to discard the changes.

4. After successfully setting the configuration, the current configuration status displays next to the **pencil** (edit) icon.

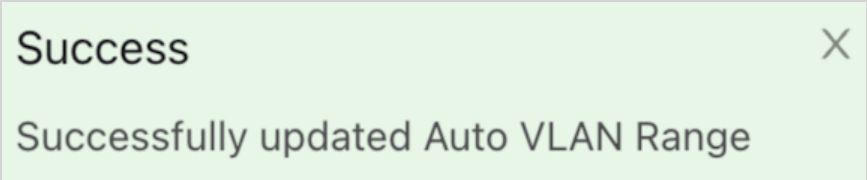
Figure 1-47: PTP Timestamping Configuration Change



Feature Setting Notification Message

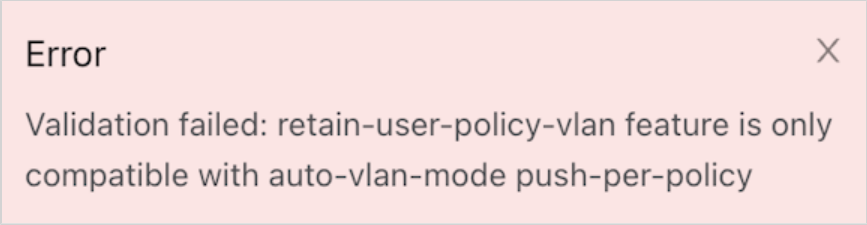
Successfully configuring a feature results in a success notification message pop-up with specific details.

Figure 1-48: Success Message



Whenever an error occurs during the configuration of a feature, an error notification message pops up along with specific details about the error.

Figure 1-49: Failure Message



Control Plane Lockdown Mode

Enable or disable the Control Plane Lockdown Mode feature.

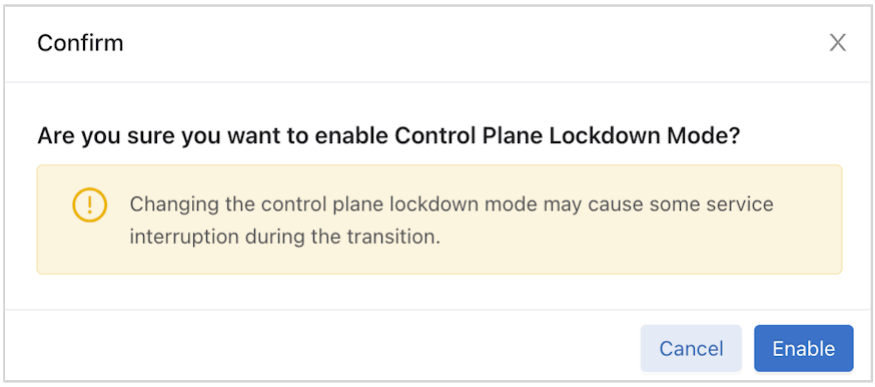
1. A **toggle button** controls the configuration of this feature. Locate the corresponding card and select the **toggle** switch.

Figure 1-50: Control Plane Lockdown Mode



2. Select **Enable** to enable Control Plane Lockdown Mode or **Cancel** to discard the changes.

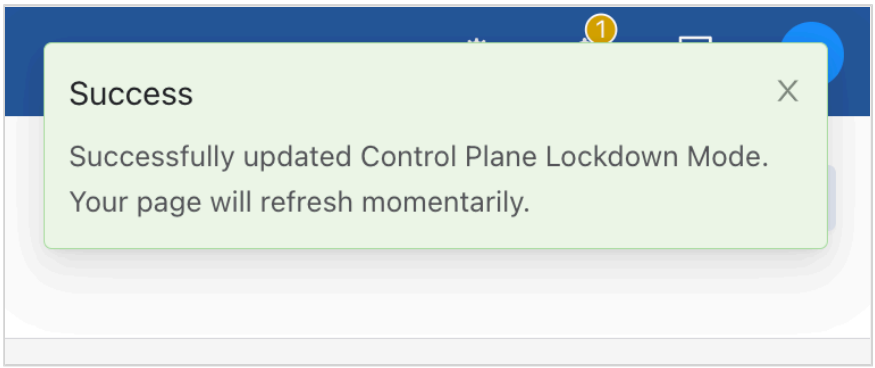
Figure 1-51: Enable Control Plane Lockdown Mode



Note: Changing the Control Plane Lockdown Mode may cause some service interruption during the transition.

3. On enabling Control Plane Lockdown Mode, a success notification message pops up with specific details.

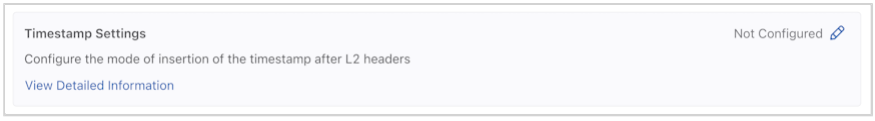
Figure 1-52: Success Message



Timestamp Settings

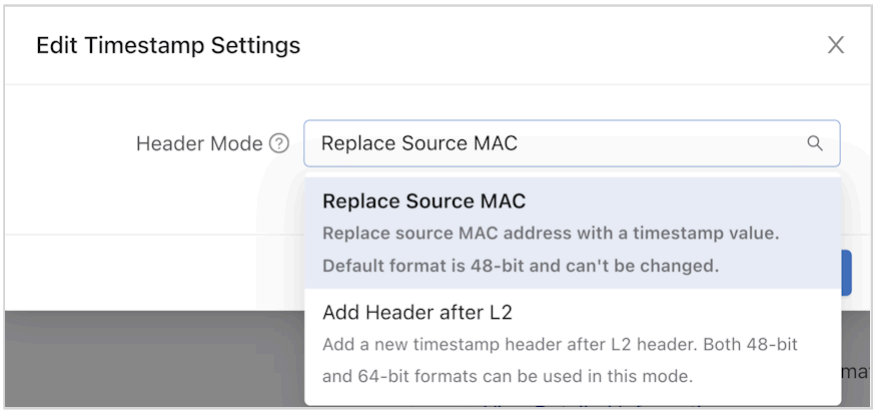
1. Control the configuration of this feature using the **Edit** icon by locating the corresponding card and selecting the **pencil** icon.

Figure 1-53: Timestamp Settings



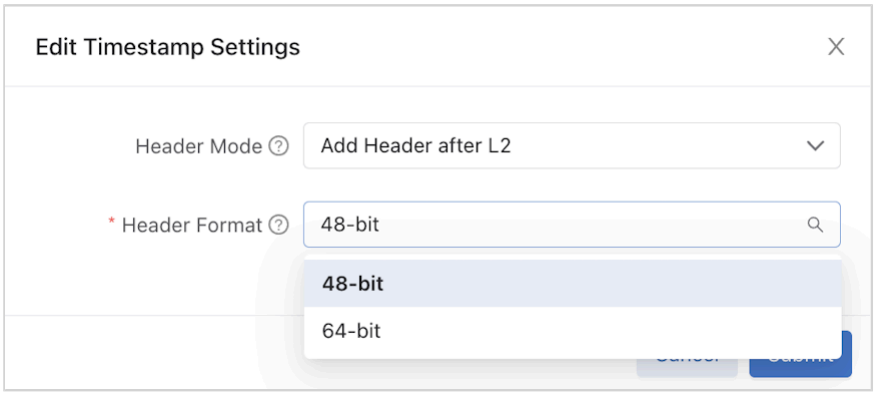
2. To configure different header modes, select the drop-down arrow. There are two ways to edit the timestamp settings - **Replace Source MAC** or **Add Header after L2**.

Figure 1-54: Edit Timestamp Settings



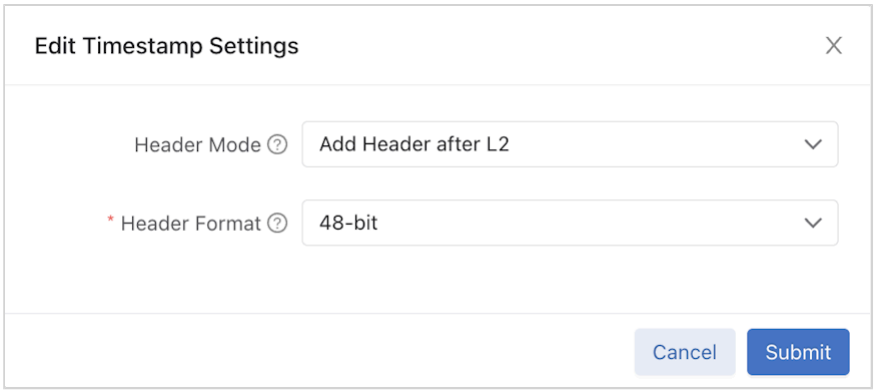
3. For **Add Header after L2** Mode, choose the header format as 48-bit or 64-bit.

Figure 1-55: Add Header after L2



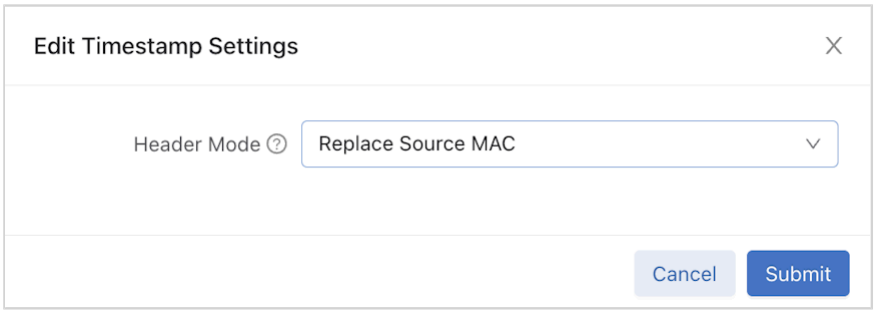
4. Select **Submit** to confirm the configuration changes or **Cancel** to discard the changes.

Figure 1-56: Submit Timestamp Changes



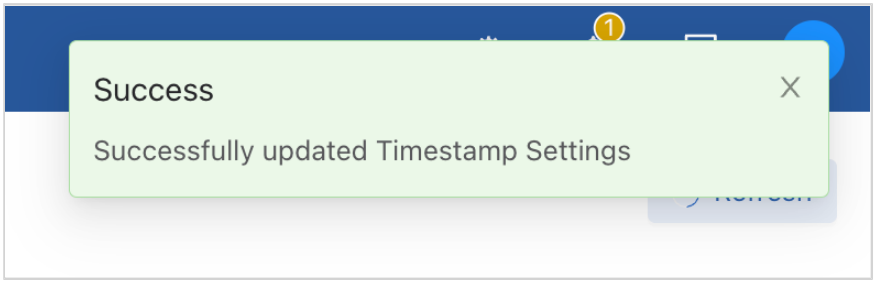
5. Select **Replace Source MAC** Mode and use **Submit** to confirm the configuration changes or **Cancel** to discard the changes.

   Figure 1-57: Replace Source MAC



6. On enabling Timestamp Settings, a success notification message pops up with specific details.

   Figure 1-58: Success Message



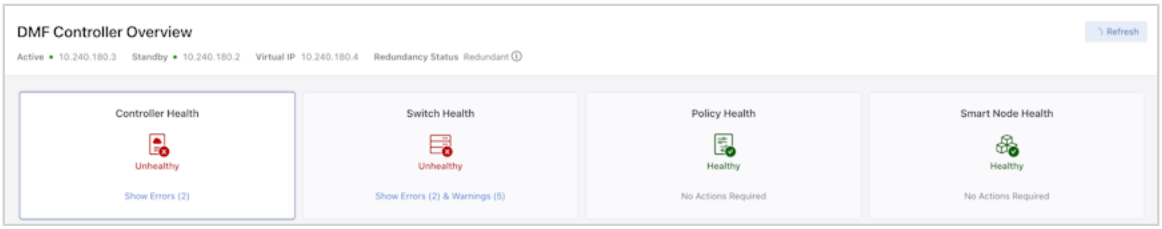
1.2.3 Dashboard Layout

The dashboard data displays four tabs:

- Controller Health
- Switch Health
- Policy Health
- Smart Node Health

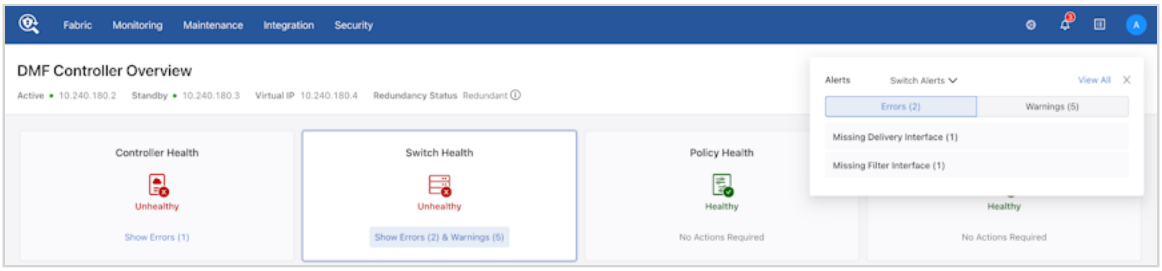
Each tab has health indicators for that category, and accessing the tab displays the relevant data below.

Figure 1-59: DANZ Monitoring Fabric (DMF) Controller Tabs



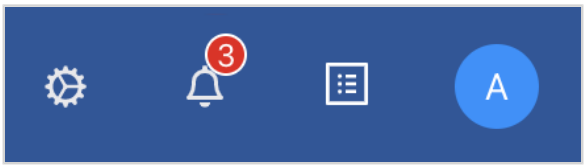
If a category contains errors or warnings, selecting the message in the tab opens a details window. It displays the number of errors or warnings filtered by tab category.

Figure 1-60: Filtered by Category



Review errors by selecting the bell icon on the right side of the Navigation bar, and it will list all fabric errors and warnings instead of filtering by an individual tab.

Figure 1-61: Notification Bell



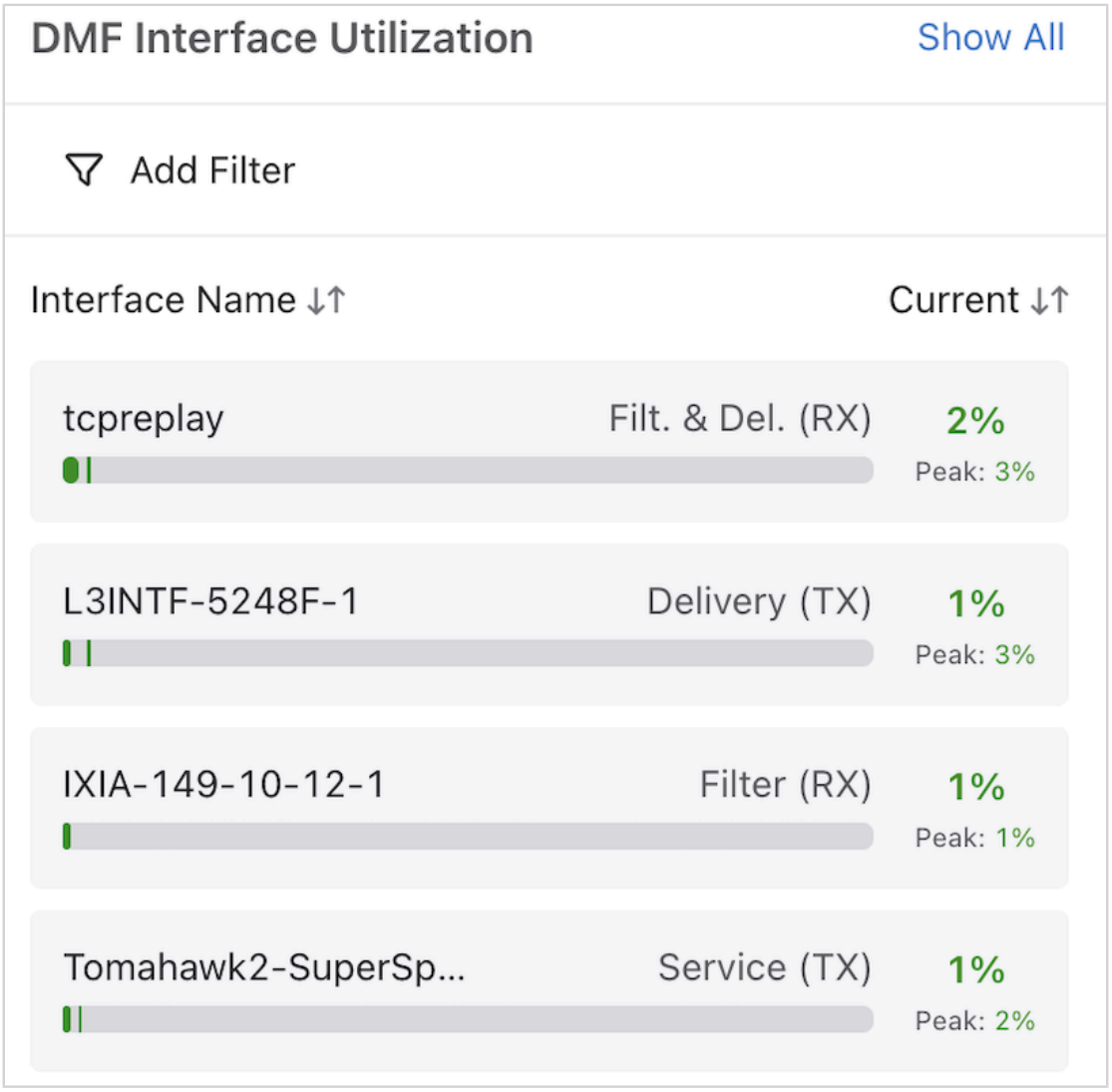
1.2.4 Controller Health

DANZ Monitoring Fabric (DMF) Interface Utilization

This widget displays the utilization of each DMF interface as follows:

- DMF Interface Name
- Interface Role
- Traffic Direction
- Current Utilization (%)
- Peak Utilization (%)

Figure 1-62: DMF Interface Utilization



The bar indicates the current utilization and shows peak utilization with a vertical line. The color of the bar and percentage changes depending on the utilization:

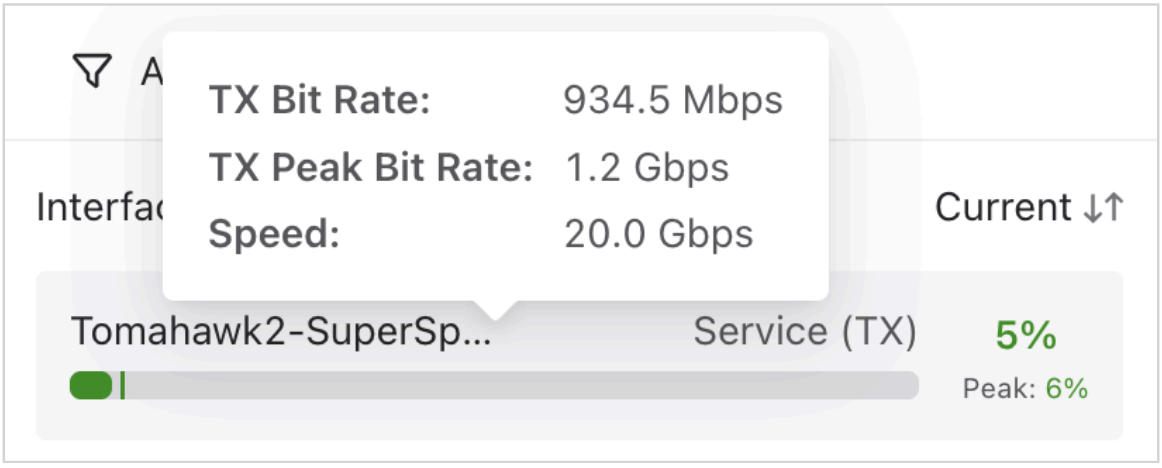
- Red means the utilization percentage is greater than 95%.
- Yellow means the utilization percentage is greater than 70%.
- Green means the utilization percentage is less than 70%.

Filter interfaces display only RX traffic, while delivery interfaces display only TX traffic. Other roles with bidirectional data can have one item for each direction of traffic, RX, or TX.

Show All leads to the DMF Interfaces page.

On hover, the bar shows the interface's Bit Rate, Peak Bit Rate, and Speed in bits per second.

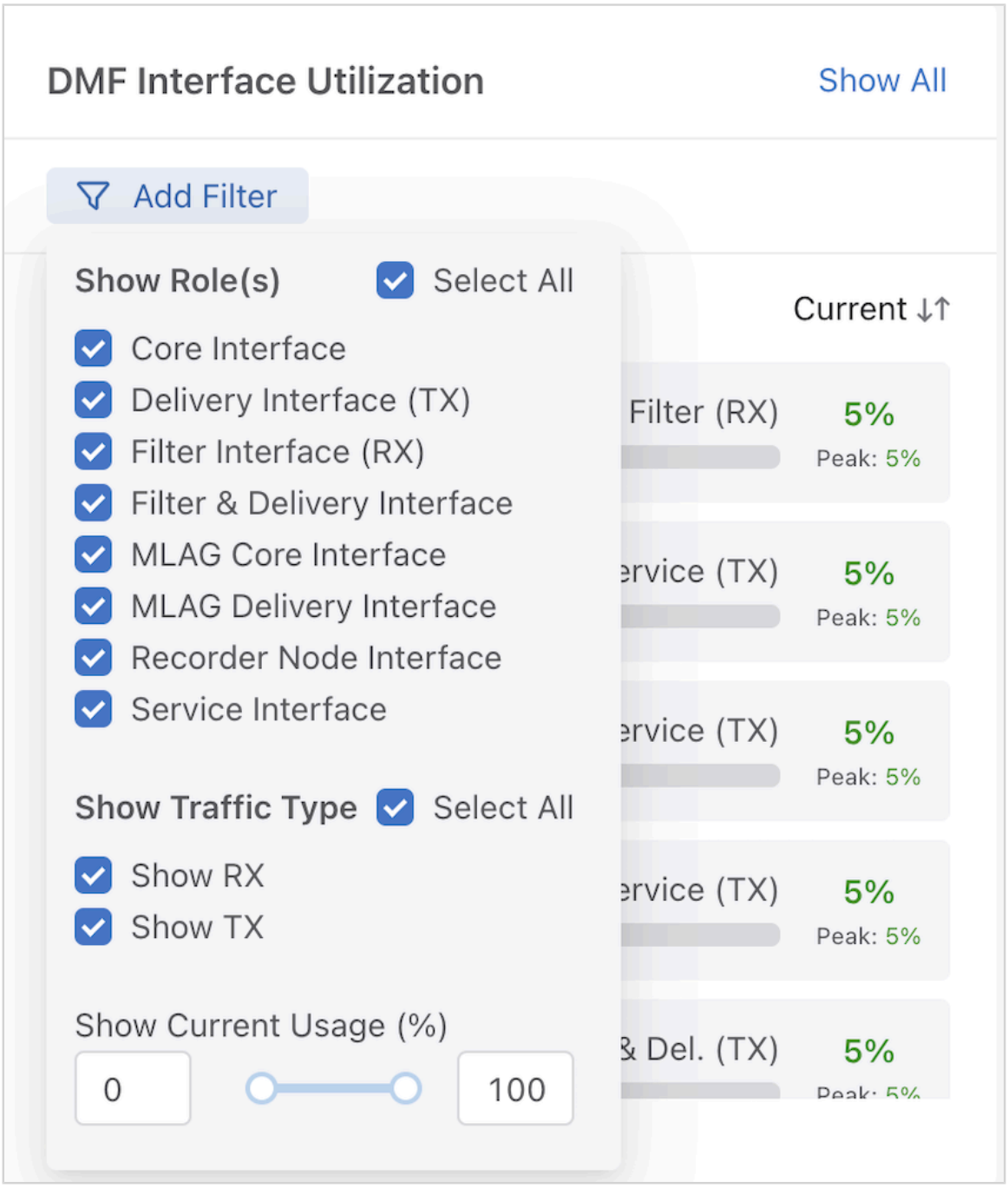
Figure 1-63: DMF Interface Utilization Hover Details



Sort interfaces by Interface Name or Current Utilization. The interfaces are sorted by current utilization (descending order) by default.

Display the interfaces by filtering using Role, Traffic Direction, and Current Utilization.

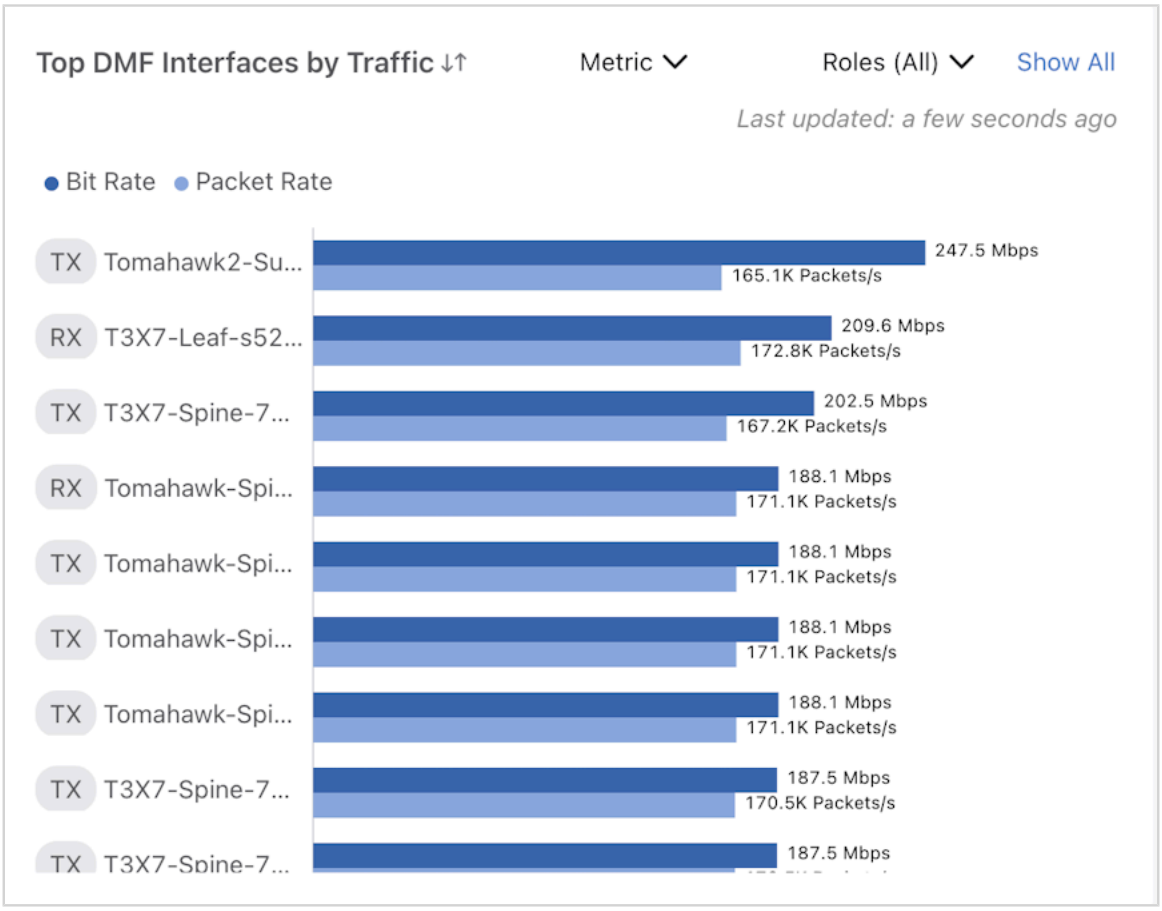
Figure 1-64: Sort Roles



Top DMF Interfaces by Traffic

This visualization displays each DMF interface's traffic (bit rate and packet rate).

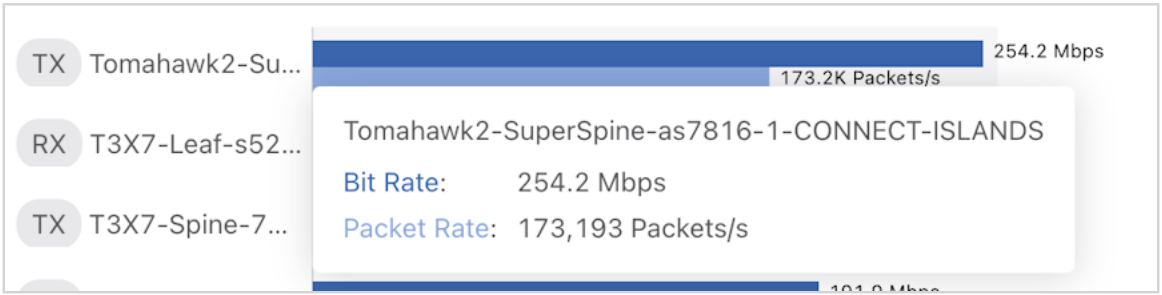
Figure 1-65: Top DMF Interfaces by Traffic



The widget shows each interface's traffic direction, DMF Interface name, bit rate, and packet rate. **Show All** leads to the DMF Interfaces page. Sort interfaces by Bit Rate and filter by Metric and Role. By default, DMF sorts the data in descending order of bit rate.

On hover, the widget shows the DMF name, bit rate, and packet rate.

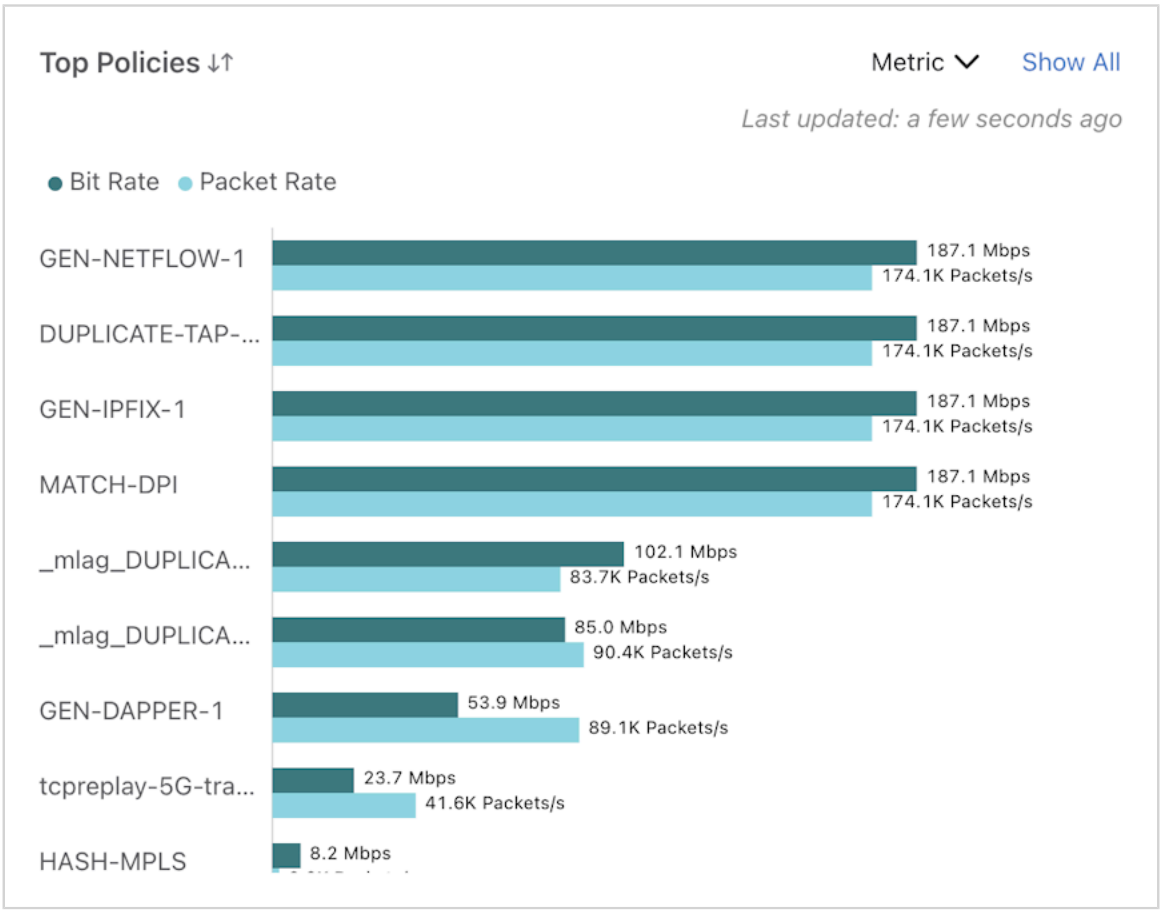
Figure 1-66: Top DMF Interfaces by Traffic Hover Details



Top Policies

The widget displays the top policies in DMF. For each policy, traffic is determined by totaling the traffic of each of its configured filter interfaces.

Figure 1-67: DMF Top Policies

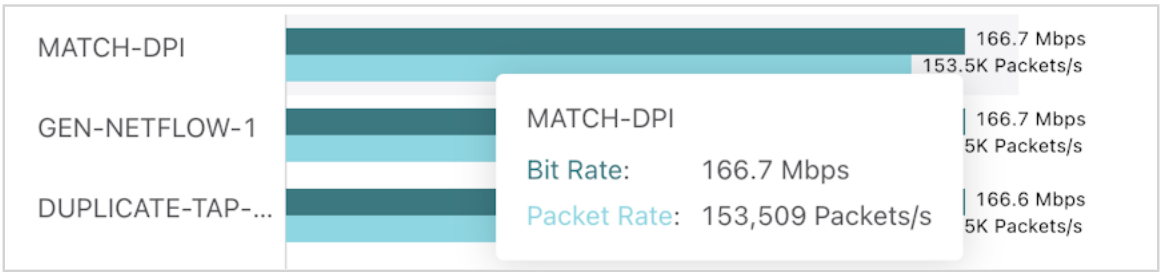


For each policy, the bar chart shows the following:

- Policy Name
- The sum of the bit rates of all filter interfaces associated with the policy.
- The sum of the packet rates of all filter interfaces associated with the policy.

On hover, the bar displays the policy name, bit rate, and packet rate.

Figure 1-68: DMF Policies Hover Details



Sort policies by Bit Rate and filter by Metric. By default, DMF sorts the policies in descending order of bit rate.

Show All leads to the Policies page.

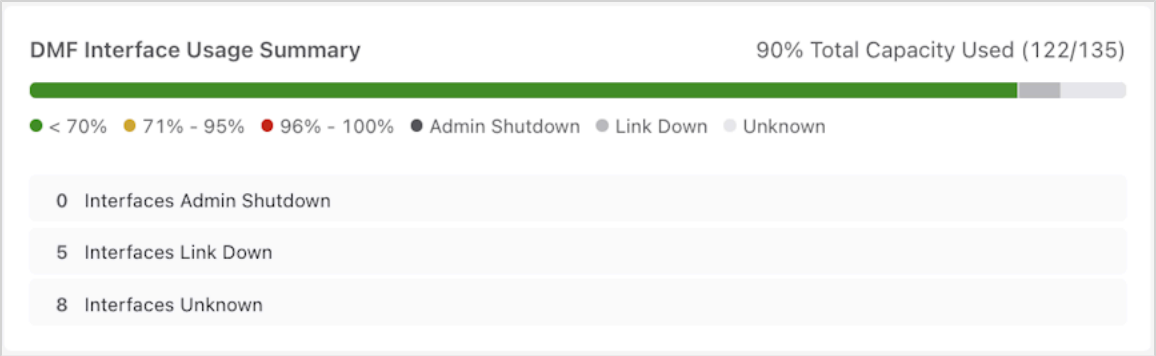
1.2.5 Switch Health

Interface Usage Summary

This widget displays the usage statistics for all DANZ Monitoring Fabric (DMF) interfaces. The interface utilization groups all active interfaces:

- Red means that the utilization percentage is greater than 95%.
- Yellow means that the utilization percentage is greater than 70%.
- Green means that the utilization percentage is less than 70%.

Figure 1-69: DMF Interface Usage Summary

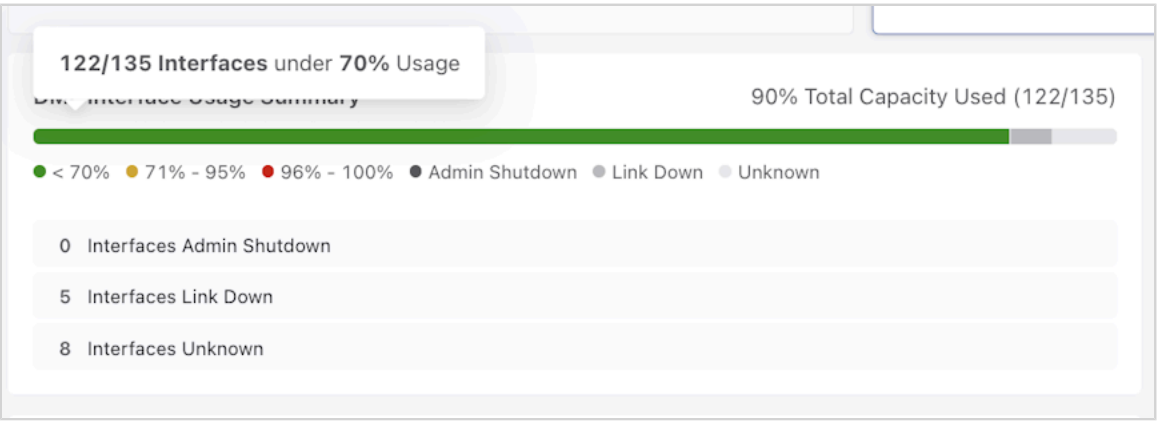


There are three other categories for DMF Interfaces with no traffic. These appear beneath the Usage Bar:

- Admin Shutdown
- Link Down
- Unknown - when Interface Speed is undefined or not known
- Total Capacity Used displays with Total Capacity defined as the number of Active DMF Interfaces divided by the Number of Total DMF Interfaces

On hover, the number of interfaces in each category appears in the respective usage bar.

Figure 1-70: DMF Interface Usage Hover Details



Switch Usage Summary

This widget displays the usage statistics for each switch. All switches are grouped by:

- Active (Green)
- Admin Shutdown (Yellow)
- Down (Red)
- Quarantined (Grey)

Figure 1-71: DMF Switch Usage Summary



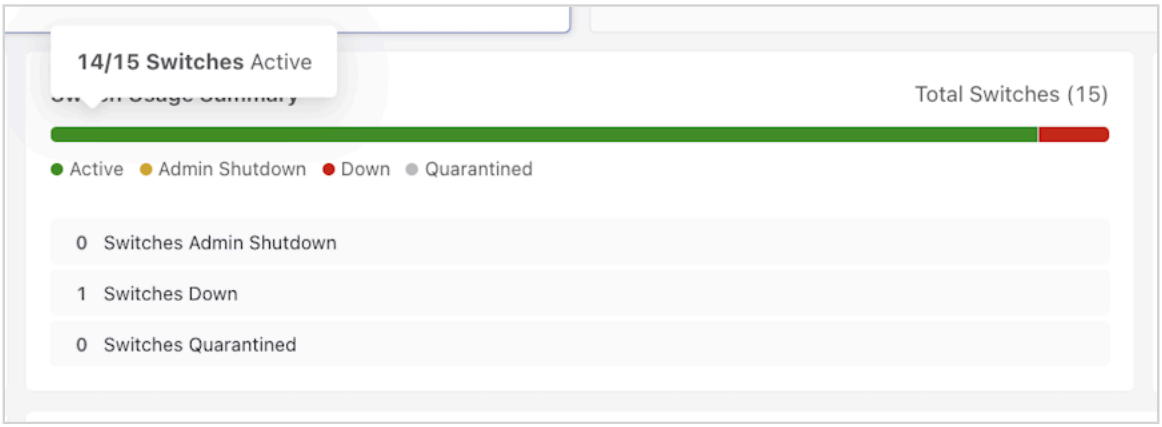
The total number of switches is displayed.

Three list items display the number of:

- Switches Admin Shutdown
- Switches Down
- Switches Quarantined

On hover, the number of switches in each category appears in the respective usage bar.

Figure 1-72: DMF Switch Usage Hover Details

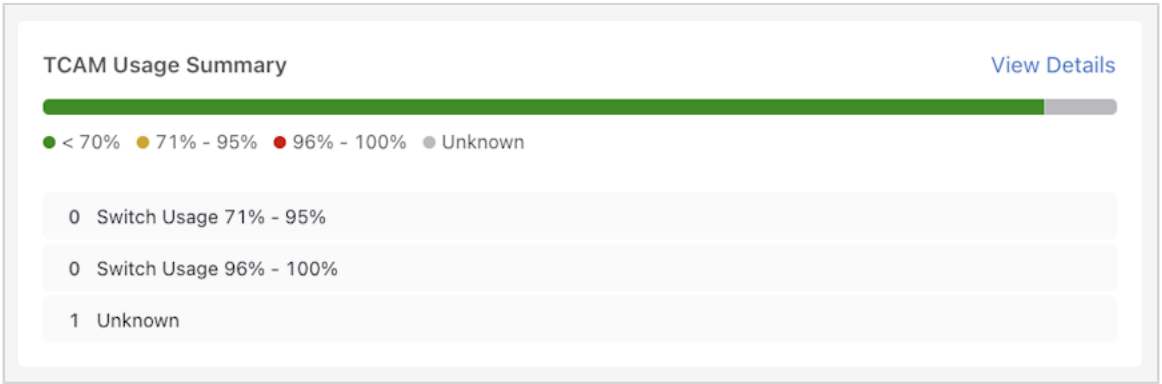


TCAM Usage Summary

This widget displays the usage statistics for the TCAM of each switch and groups all active TCAMs by usage:

- Red means that the utilization percentage is greater than 95%.
- Yellow means that the utilization percentage is greater than 70%.
- Green means that the utilization percentage is less than 70%.
- Grey means that the utilization is Unknown.
  - A switch is grouped in the Unknown category when no TCAM usage statistics are available, generally from a switch being shut down or disconnected.

Figure 1-73: DMF TCAM Usage Summary



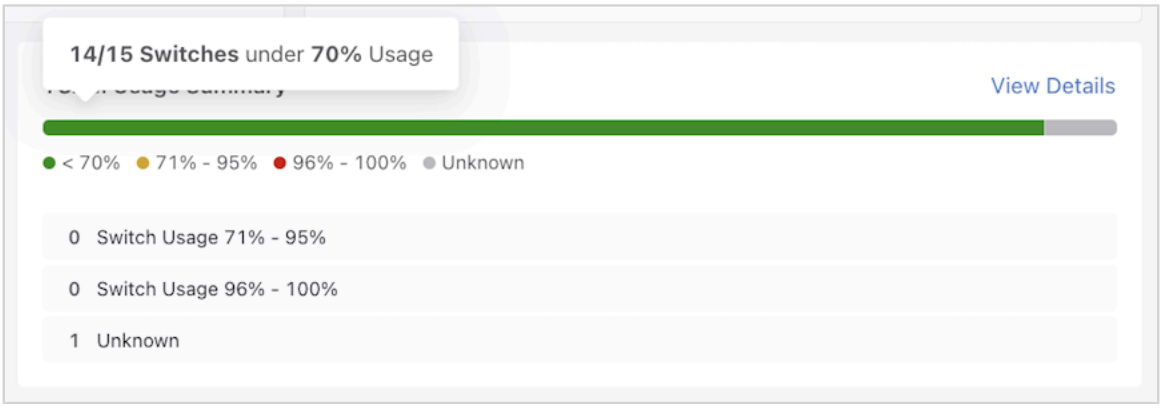
The **View Details** link leads to the TCAM Utilization tab of the Switches page.

Below the Usage Bar, there are three list items displaying:

- Switch Usage 71% - 95%
- Switch Usage 96% - 100%
- Unknown

On hover, the number of switches in each category appears in the respective usage bar.

Figure 1-74: DMF TCAM Utilization Hover Details



DMF Interface Utilization

DMF Interface Utilization is similar to the data displayed in the Controller Health tab. Please refer to its description for more information.

Switch Utilization

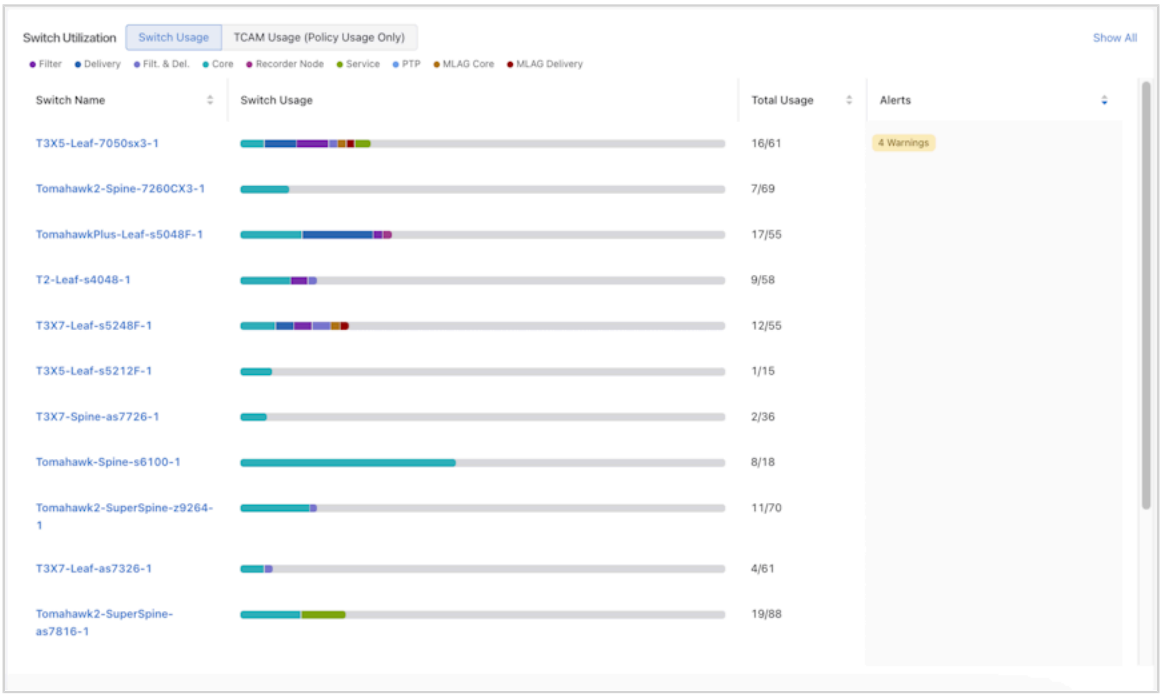
This widget contains two tabs:

- Switch Usage
- TCAM Usage

Switch Usage

The Switch Usage tab of the Switch Utilization displays essential information for each switch, including the use of each switch interface and alerts for any warnings or errors.

Figure 1-75: DMF Switch Usage Tab



The widget displays the following data for each switch:

- Switch Name (contains a link to the Switches page for that specific switch).
- Switch Usage: Each section represents the number of interfaces with a specific role.
- Total Usage: Displays the Number of Interfaces with an assigned role divided by the Total Number of Interfaces on the switch.
- Alerts: This column displays any alerts related to interfaces.
  - The yellow badge indicates the number of warnings, while the red badge shows the number of errors.

The Switch Usage column contains the number of interfaces for each role:

- Filter
- Delivery
- Filter and Delivery
- Core
- Recorder Node
- Service
- PTP

- MLAG Core
- MLAG Delivery.

Three columns can sort the table:

- Sort the Switch Name column in alphabetical order.
- Sort the Total Usage column by percentage (%) usage (# used interfaces / # total interfaces).
- Sort the Alerts column by the total number of alerts (# warnings + # errors).

The default sort order for this table is the Alerts column in descending order, which ensures the switches with the highest number of alerts are initially at the top.

On hover, the number of each interface appears.

Figure 1-76: DMF Switch Usage Filter Interfaces Hover Details



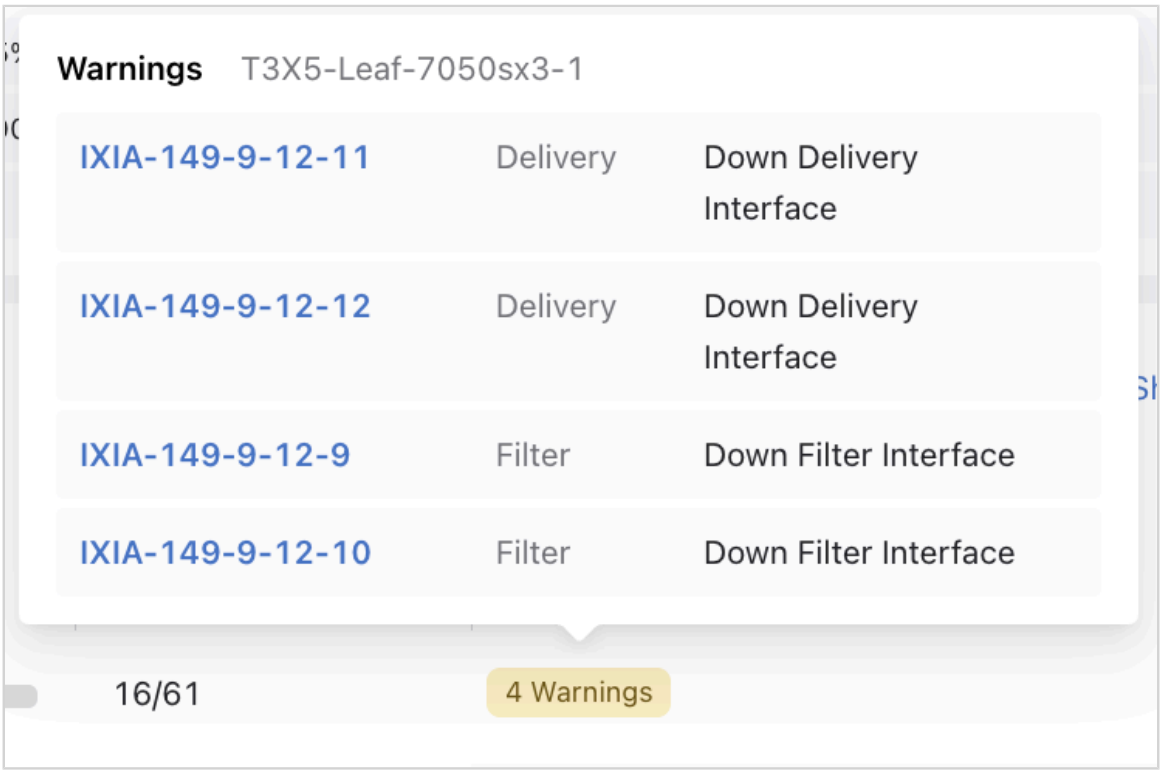
While hovering over the warnings or alerts badge, a table appears and displays Warnings for the yellow badge and Errors for the red badge, and it will also show the switch name.

Each row of the table contains the following:

- Interface name (includes a link to Interfaces/[INTERFACE-NAME] page)
- Interface role

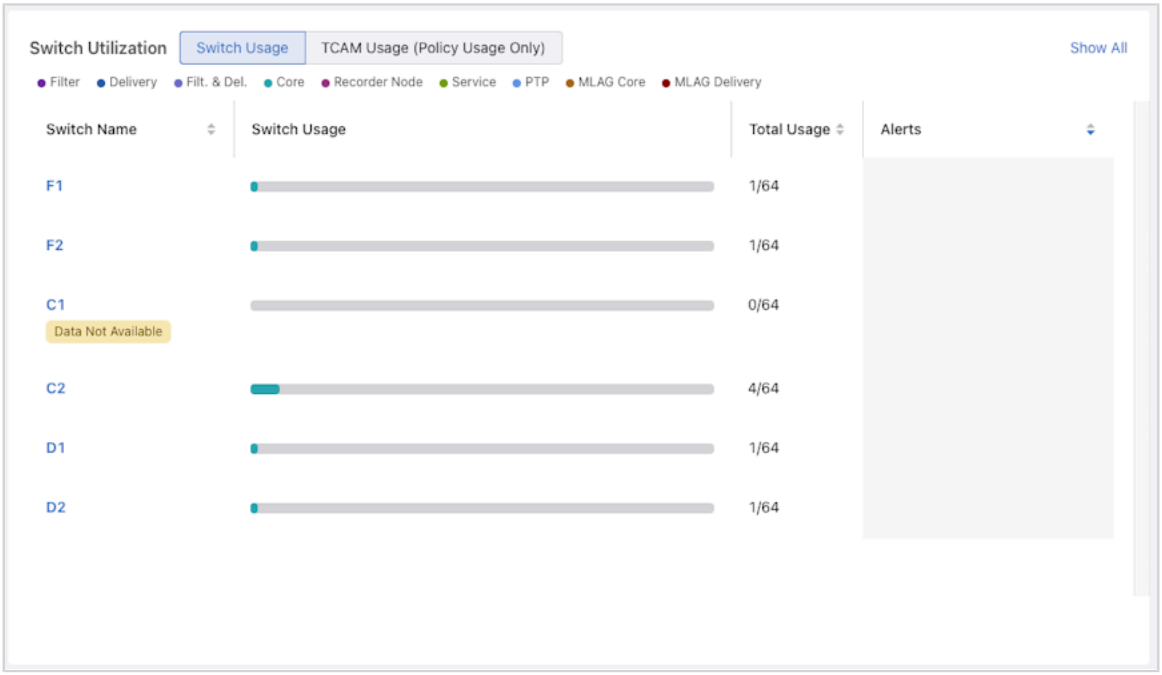
- Alert type (e.g., Down Delivery Interface)

Figure 1-77: DMF Interface Warnings



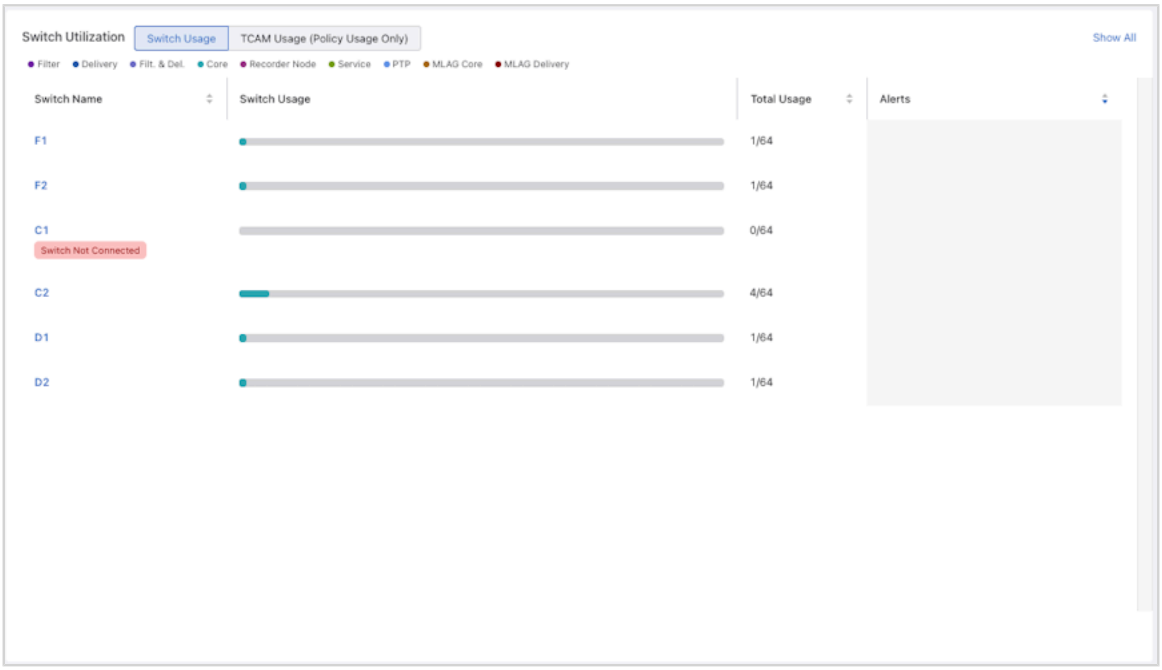
When data is unavailable for a switch (C1), there will be a yellow badge under the Switch Name that says **Data Not Available**. The Switch Usage column will have an empty usage bar, the Total Usage Column will show 0 (zero) for the number of currently used interfaces, and the Alerts column will be empty.

Figure 1-78: DMF Switch Usage - Data Not Available



When a switch is down, a red badge appears under the Switch Name that says **Switch Not Connected**. The other columns will be empty in the same way as the **Data Not Available** case.

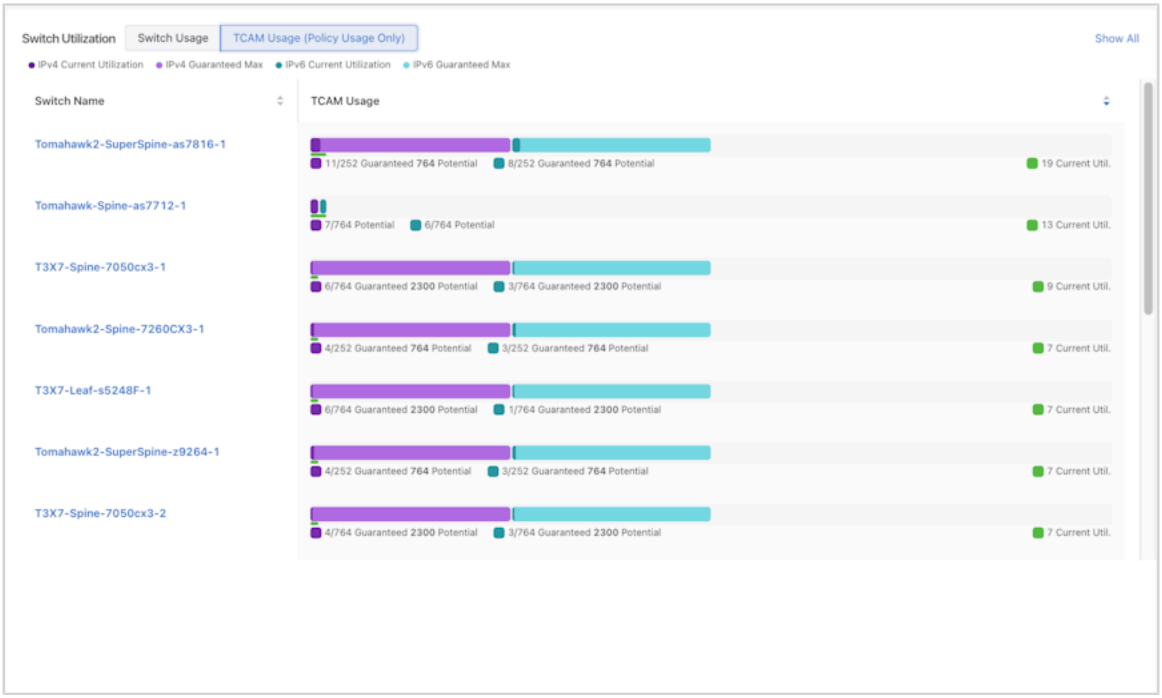
Figure 1-79: DMF Switch Usage - Switch Not Connected



TCAM Usage

The TCAM Usage widget displays the current utilization of the TCAMs for each active switch. A switch can have a TCAM for IPv4, IPv6, or both. Each TCAM has a guaranteed maximum usage and current utilization. This table compares the current utilization of each TCAM to its guaranteed maximum.

Figure 1-80: DMF Usage (Policy Usage Only)



This widget displays a TCAM Usage chart for each switch:

- The purple bar shows IPv4 Current Utilization and Guaranteed Maximum.
- The cyan bar shows IPv6 Current Utilization and Guaranteed Maximum.
- Each row will display Current Utilization (IPv4 + IPv6 Current Utilization)
- Sort by Switch Name and Current Utilization.
- Sort the Switch Name column alphabetically (descending and ascending).
- Sort the Current Utilization column in descending and ascending order (IPv4 + IPv6 Current Utilization).
- The default sort order for the table is the Current Utilization column in descending order, ensuring the switches with the highest current utilization display first.

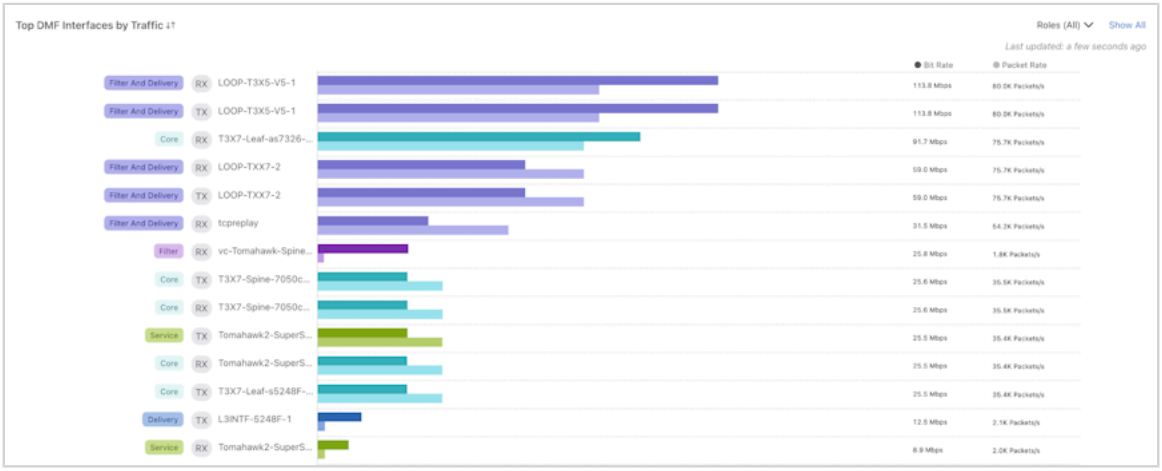
Top DMF Interfaces by Traffic

The visualization shows DMF interface traffic (bit rate and packet rate) color-coded by interface role. The roles displayed are:

- Core
- Delivery
- Filter
- Filter and Delivery

- MLAG Core
- MLAG Delivery
- Recorder Node
- Service

Figure 1-81: DMF Top Interfaces by Traffic



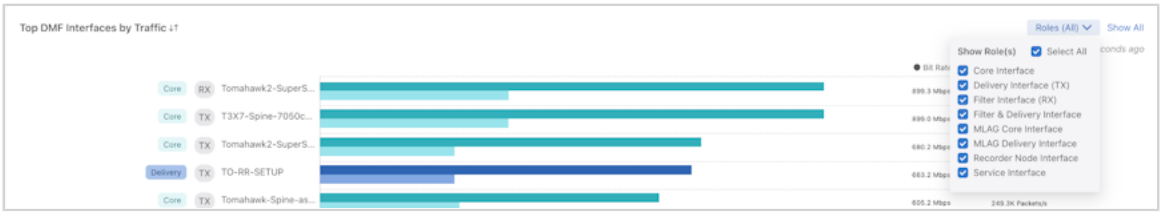
For each interface, the chart item shows:

- Interface role
- Traffic direction
- DMF interface name
- Bit rate
- Packet rate

Show All leads to the DMF Interfaces page.

Sort the interfaces by bit rate, which, by default, are sorted in descending bit rate order. Filter interfaces by interface role using the drop-down.

Figure 1-82: DMF Sort Interfaces by Bit Rate



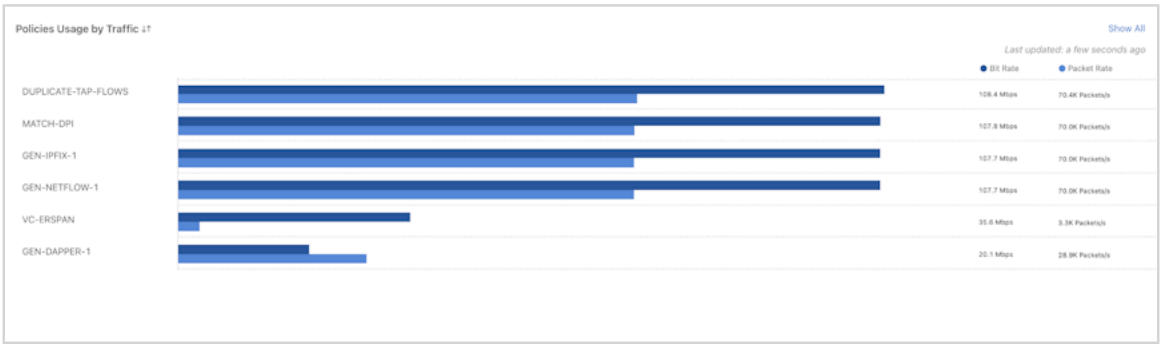
1.2.6 Policy Health

Policies Usage by Traffic

This widget displays policy traffic. For each policy, the bar chart shows:

- Name of the policy
- Bit rate
- Packet rate

Figure 1-83: DANZ Monitoring Fabric (DMF) Policies Usage by Traffic



On hover, similar information displays.

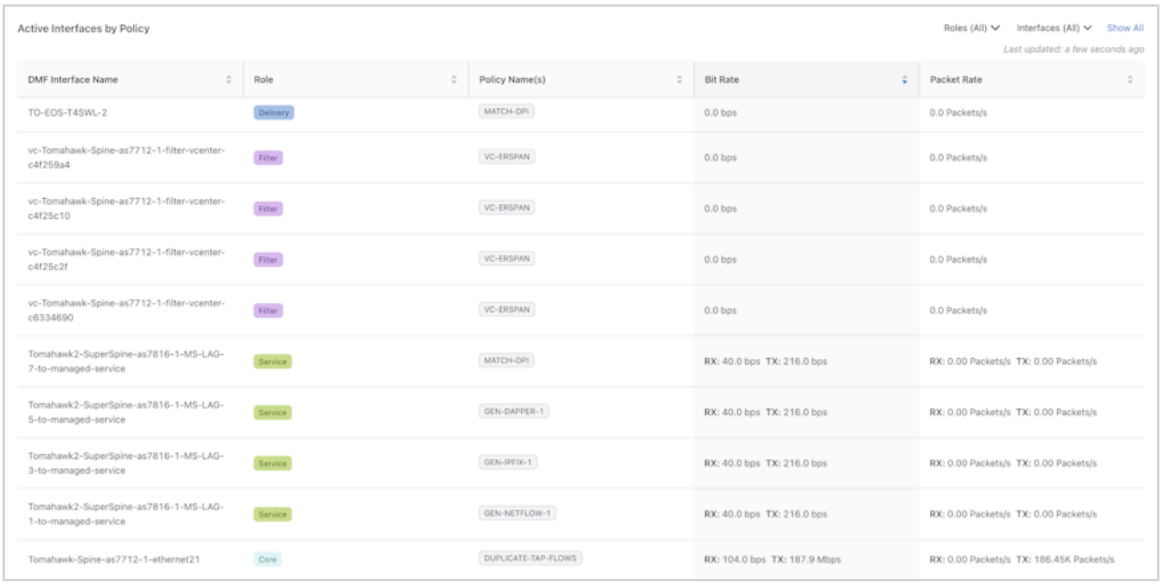
Sort policies by Bit Rate.

Show All leads to the Policies page.

Active Interfaces by Policy

The table displays DMF interfaces associated with policies. DMF interfaces that are not affiliated with a policy are not displayed.

Figure 1-84: DMF Active Interfaces by Policy



The table contains the following columns:

- DMF Interface Name: The DMF name of the switch interface.
- Role: The role of the interface.
- Policy Name(s): A list of the policies associated with the interface.
- Bit Rate: The bit rate of the interface.
- Packet Rate: The packet rate of the interface.

Show All leads to the DMF Interfaces page.

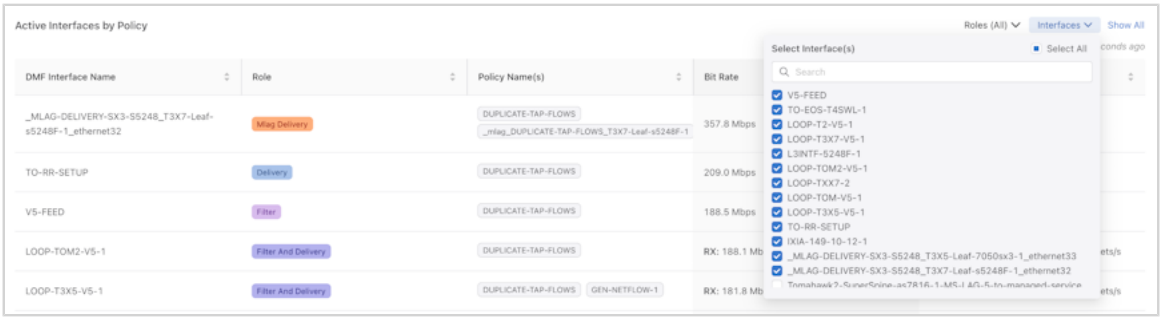
Sort the table by each column; DMF sorts the items in descending bit rate order by default.

Two filters, **Roles** and **Interfaces**, allow data sorting by interface role and DMF interface name.

Figure 1-85: DMF Active Interfaces by Policy - Roles



Figure 1-86: DMF Active Interfaces by Policy - Interfaces



1.2.7 Smart Node Health

Recorder Node

The Recorder Nodes table displays Recorder Node health and the following columns:

- Recorder Node Name
- IP Address
- MAC Address
- Recording
  - Indicates the status of the Recorder Node recording configuration, either `Yes` or `No`.
- Storage Utilization
- Index and Packet disk storage utilization % (percentage) using the following colors:
  - Red means the utilization percentage is greater than 95%.
  - Yellow means the utilization percentage is greater than 70%.
  - Green means the utilization percentage is less than 70%.

Figure 1-87: Recorder Nodes



On hover, various details appear depending on the column selected. These include:

- Free and Total Disk Usage
- Backup Storage Utilization
- Index and Packet backup disk storage utilization % (percentage) using the following colors.
  - Red means the utilization percentage is greater than 95%.
  - Yellow means the utilization percentage is greater than 70%.
  - Green means the utilization percentage is less than 70%.
- Virtual Disk Health
- Status of Index and Packet virtual disks:
  - Green means the virtual disk's health is good.
  - >Red means the value of the virtual disk's health is bad.
- Recorder Node Fabric Interface
  - Shows the DMF interface name and its status where the Recorder Node connects to the DMF Fabric.

- Switch, Interface, and status
- Zero Touch State
- Alerts
- Errors and warnings for the Recorder Node - Hovering over an error displays additional information about the errors and warnings.

The following are examples of detailed information when hovering.

Figure 1-88: Example - Index Disk Storage

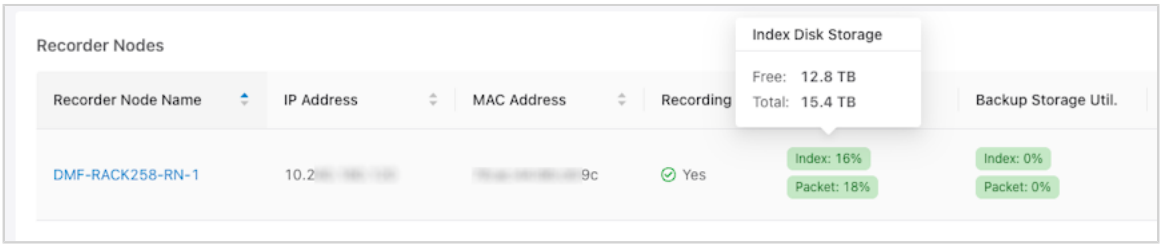


Figure 1-89: Example - Index Backup Disk Storage

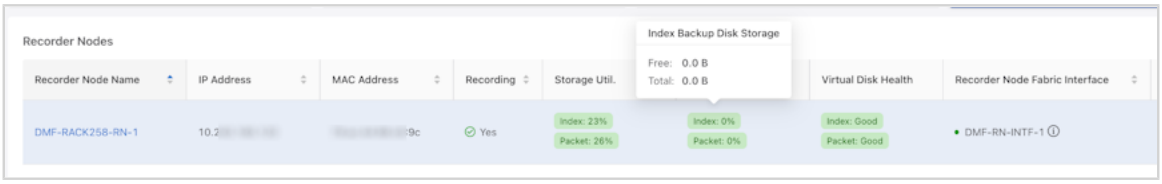


Figure 1-90: Example - Recorder Node Fabric Interface

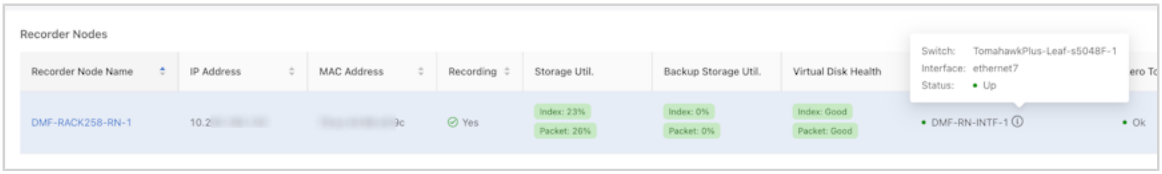


Figure 1-91: Example - Errors



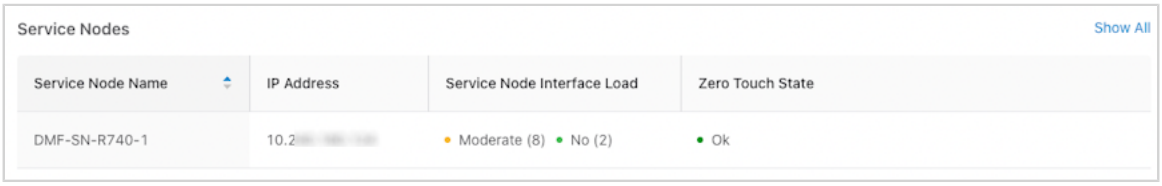
The **View All** link leads to the **Recorder Node** page.

Service Node

The Service Nodes table displays Service Node health and the following columns:

- Service Node Name
- IP Address
- Service Node Interface Load
- Zero Touch State

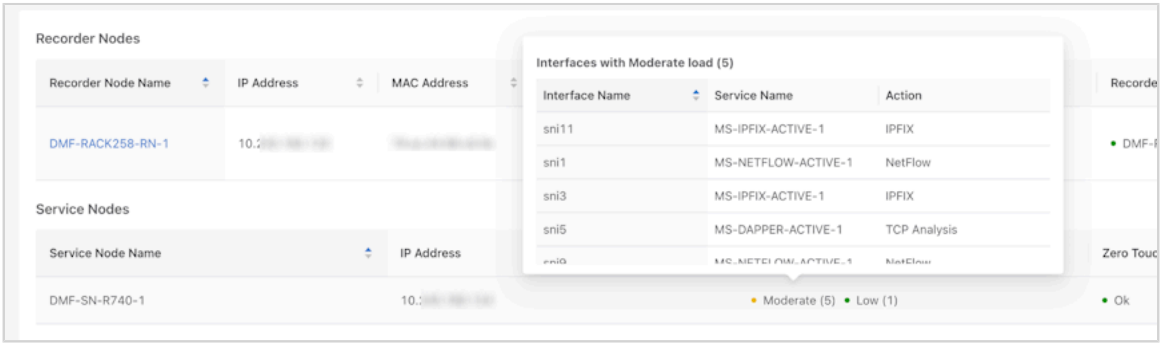
Figure 1-92: DANZ Monitoring Fabric (DMF) Service Nodes



Hovering over the Service Node Interface Load column displays:

- Interface Name
- Service Name
- Action

Figure 1-93: DMF Service Nodes Hover Details



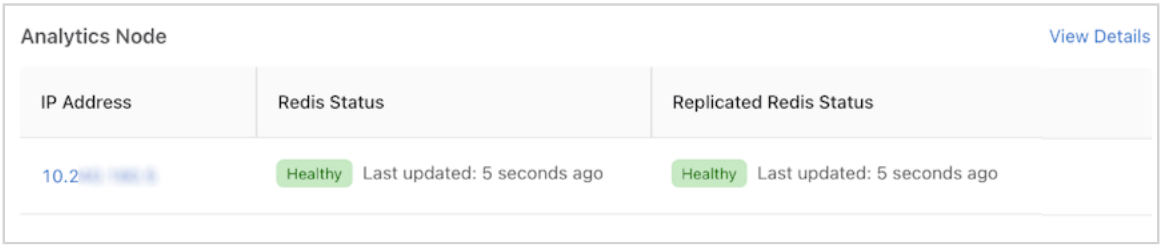
The **View All** link leads to the **Service Node** page.

Analytics Node

The Analytics Node table displays Analytics Node health and the following columns:

- IP Address: The configured Analytics Node IP address.
  - Selecting the **IP Address** opens the Analytics Node UI.
- Redis Status
  - Displays the status in green if healthy, along with the last updated timestamp.
  - Displays the status in red if unhealthy, along with the latest updated timestamp.
- Replicated Redis Status
  - Displays the status in green if healthy, along with the latest updated timestamp.
  - Displays the status in red if unhealthy, along with the latest updated timestamp

Figure 1-94: DMF Analytics Node



The **View Details** link leads to the **Analytics Node** details page.

Refreshing Data

Data automatically refreshes every minute, and interface topology data automatically refreshes every 10 seconds.

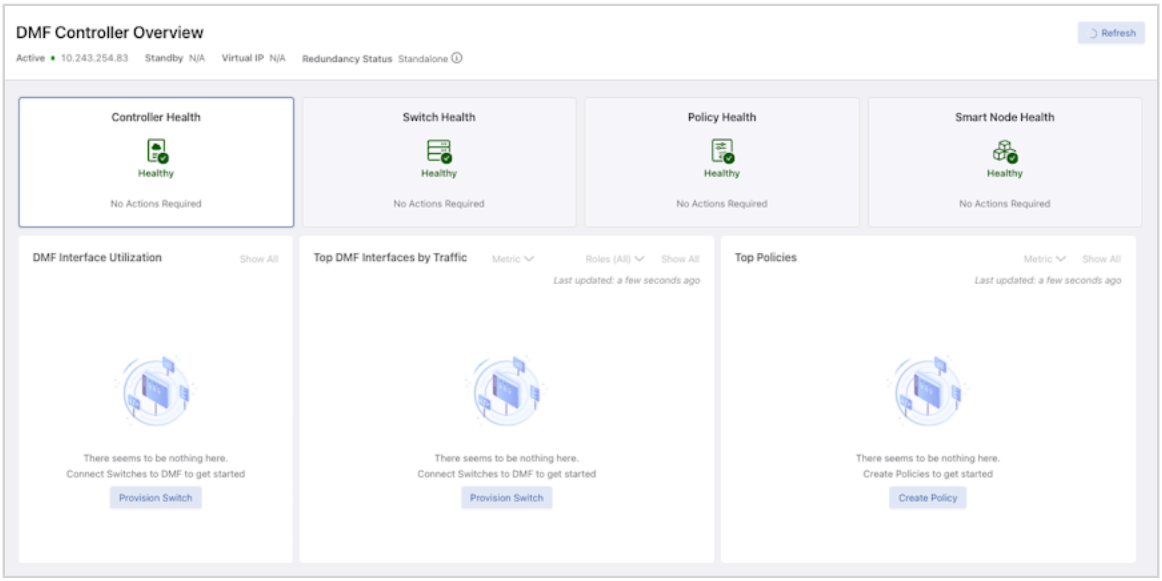
Manually refresh dashboard data using **Refresh**.

1.2.8 Empty State

When there are no provisioned switches, DANZ Monitoring Fabric (DMF) Interface Utilization and Top DMF Interfaces by Traffic will display an Empty Component.

Each empty component contains a link to provision a switch. The system prompts the user to create a DMF interface if there are provisioned switches but no assigned DMF interfaces.

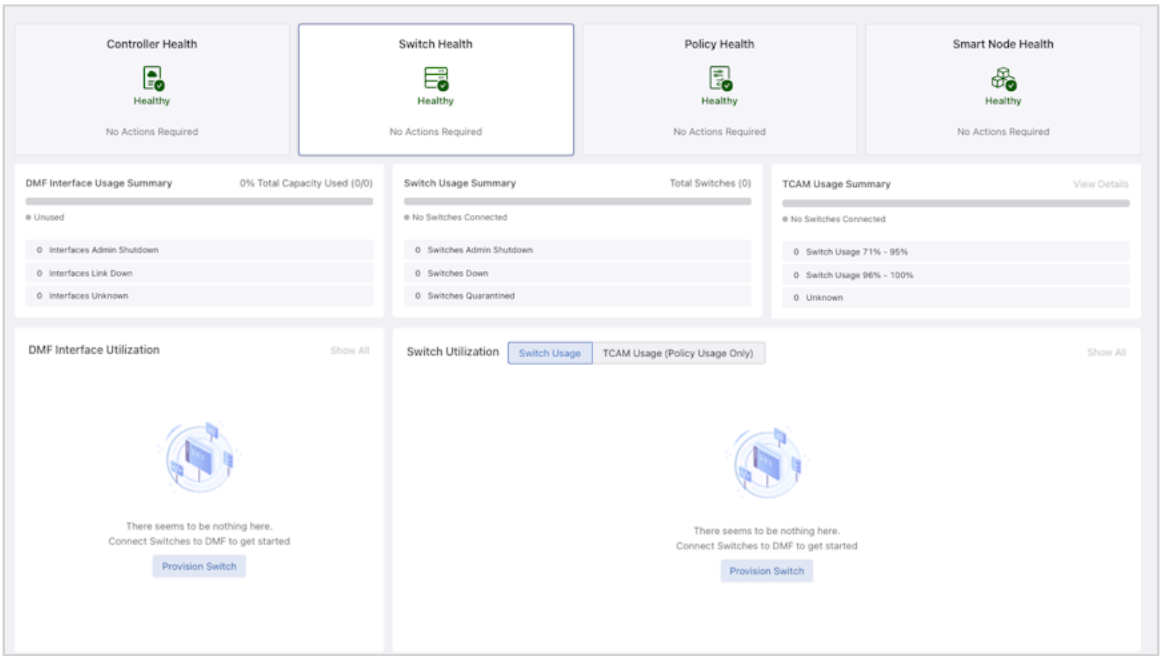
Figure 1-95: DMF Controller Overview - Empty State



Top Policies will display an Empty Component if no policies exist.

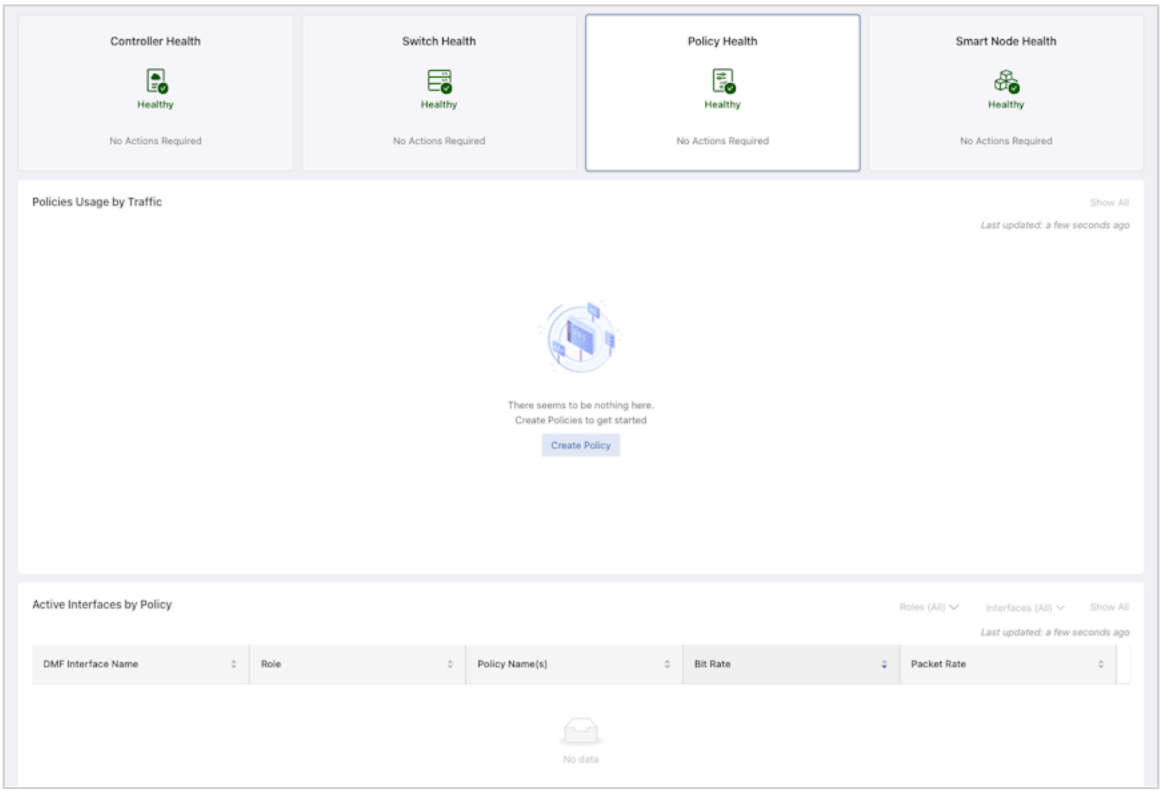
Select **Create Policy** to go to the **Create Policy** page.

Figure 1-96: DMF Switch Health - Empty State



The Usage Summary components will display Unused or No Switches Connected for the usage bar legend.

Figure 1-97: DMF Policy Health - Empty State



Policies Usage by Traffic displays the same Empty Component as Top Policies.

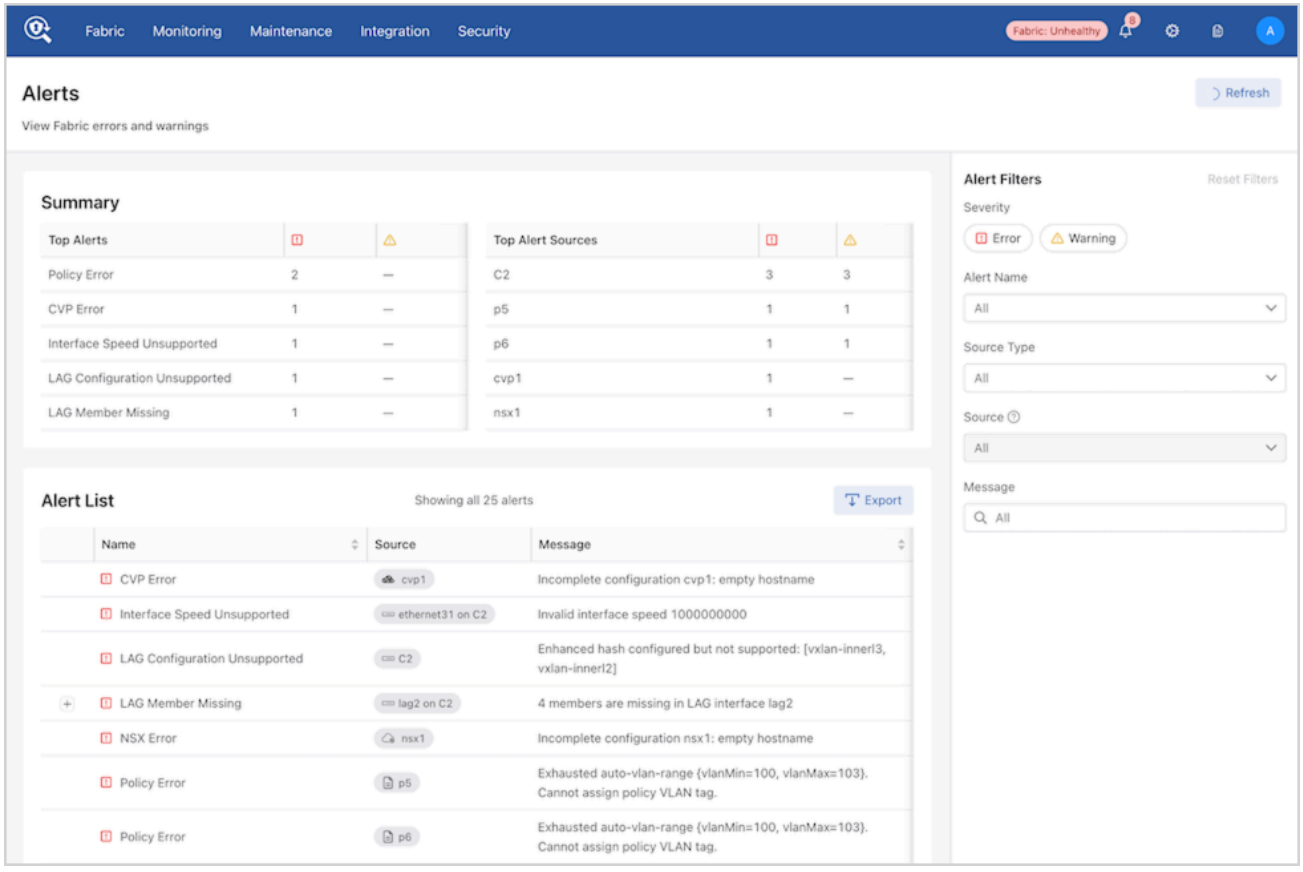
1.2.9 Alerts

The **Alerts** page is the central hub for monitoring system errors and warnings. It provides a quick summary of critical issues and a detailed list of all active alerts, enabling swift identification, investigation, and subsequent addressing of potential problems within the fabric.

Key Features:

- **Enhanced Visibility:** A clear summary section at the top provides an immediate snapshot of the most critical alerts.
- **Intuitive Filtering:** A dedicated **Alert Filters** panel makes it easier to narrow down alerts based on severity, name, source, and message.
- **Detailed Alert List:** The **Alert List** offers a comprehensive view with sortable columns and expandable details for each alert.
- **Improved User Experience:** Streamlined navigation and a clean layout ensure a more efficient workflow.

Figure 1-98: Alerts Dashboard



Navigation

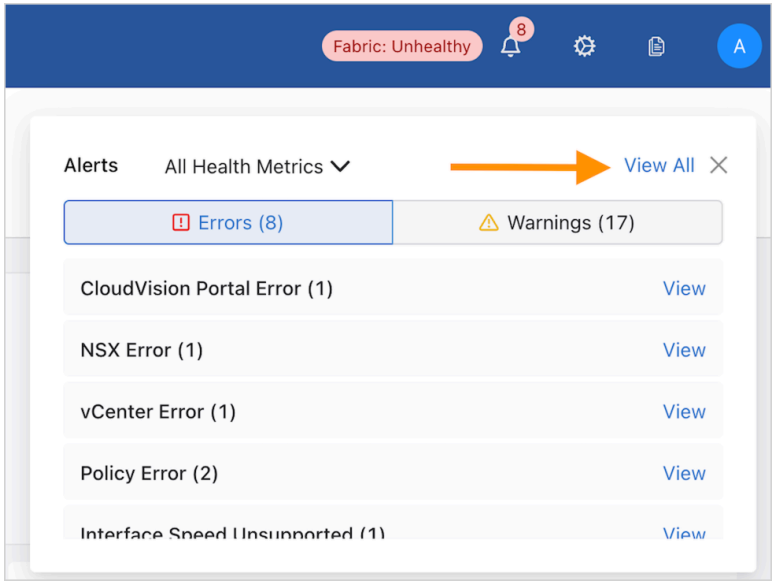
Navigate to the **Alerts** page by selecting the **notification icon** in the navigation bar.

Figure 1-99: Notification Bell



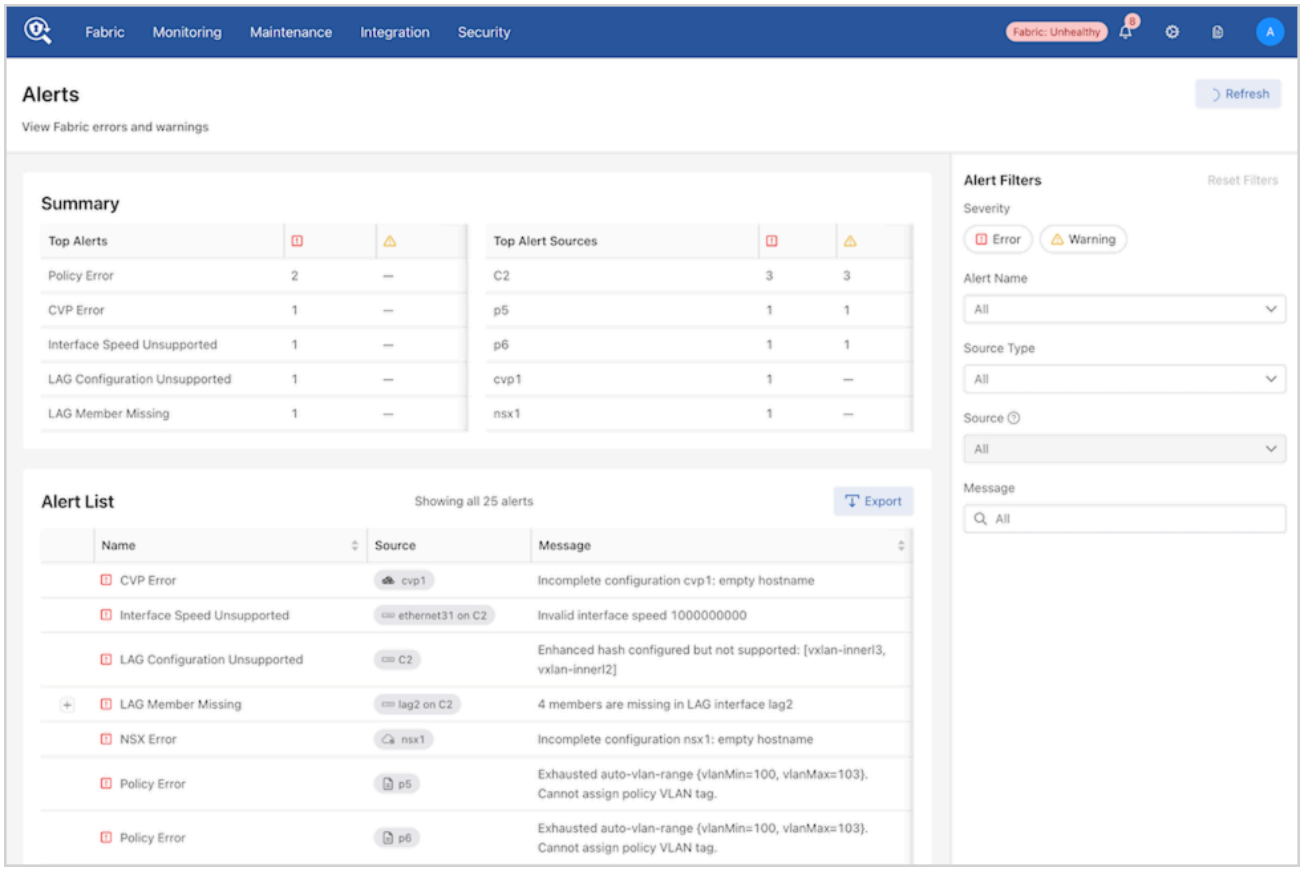
DMF displays a summary of fabric alerts. Select **View All** to proceed to the Alerts page.

Figure 1-100: View All



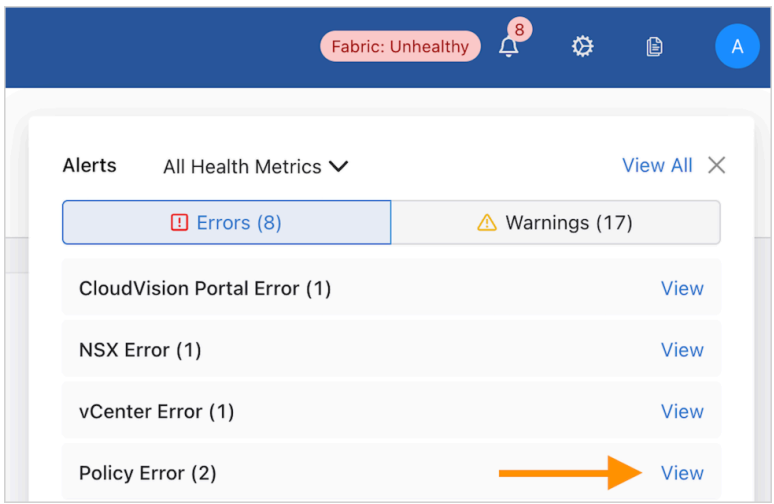
The **Alerts** dashboard displays the current list of Alerts and Warnings.

Figure 1-101: Alerts Dashboard



Select **View** to display a specific alert.

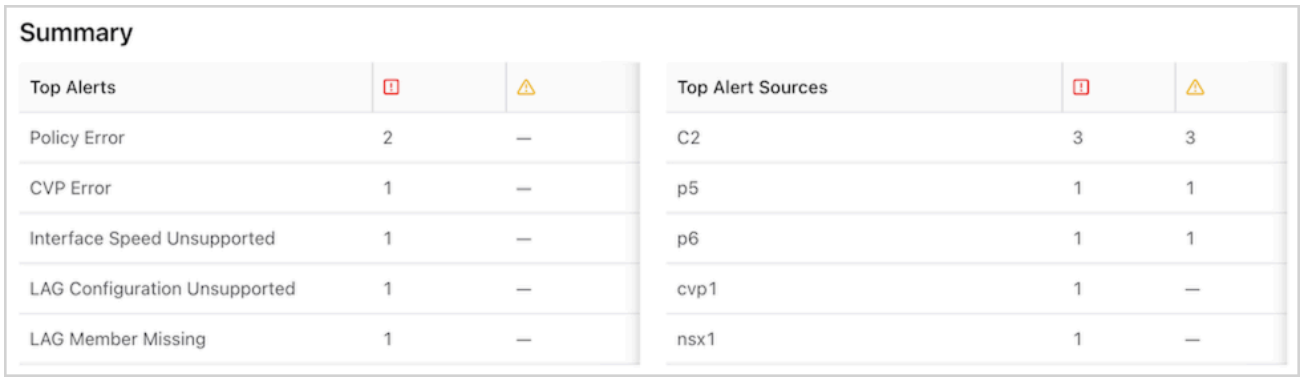
Figure 1-102: View



Summary

The **Summary** section provides a high-level overview of the most frequent and impactful alerts.

Figure 1-103: Summary



The Summary section contains two key areas:

Top Alerts

Top Alerts displays the top 5 names of the current and most prevalent alerts and the count of active instances for each:

- **Total**: The total number of active alerts for that specific alert name.
- **Error** (Red Square): The count of alerts categorized as **Error** severity.
- **Warning** (Yellow Triangle): The count of alerts categorized as **Warning** severity.

> **Note:** Alert names are ranked by severity and then by count.

Top Alert Sources

Top Alert Sources highlights the top 5 sources (e.g., specific devices and policies) generating the most alerts. Similar to **Top Alerts**, it shows:

- **Total**: The total number of alerts originating from that specific source.
- **Error** (Red Square): The count of alerts categorized as **Error** severity from that source.
- **Warning** (Yellow Triangle): The count of alerts categorized as **Warning** severity from that source.

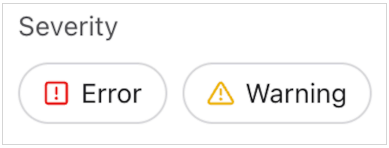
> **Note:** Alert names are ranked by severity and then by count. This summary allows for quick identification of areas of concern and prioritization of investigation.

Alert Filters

The **Alert Filters** section is a powerful tool for refining alert views. Combine multiple filters to pinpoint specific issues. When filters are applied, DMF updates both the **Summary** and **Alert List** sections.

- **Severity**
  - **Error**: Displays only alerts categorized as errors.
  - **Warning**: Displays only alerts categorized as warnings.

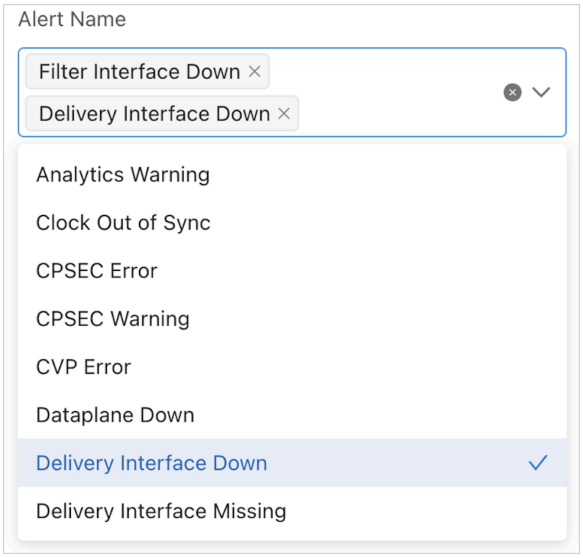
    Figure 1-104: Severity



> **Note:** Select or deselect both **Error** and **Warning** to view all alerts.

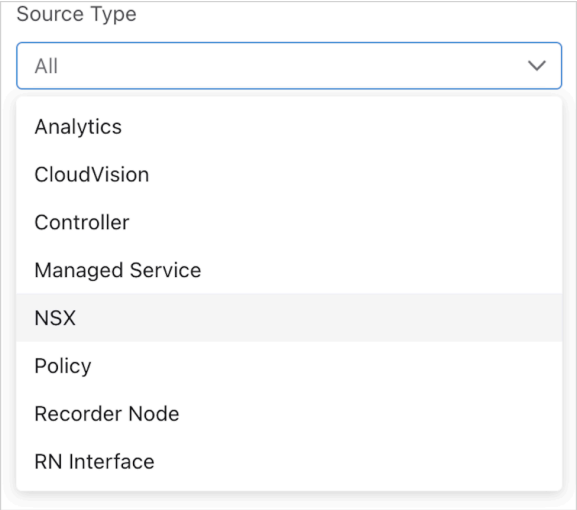
- **Alert Name**: Use the drop-down menu to select a **specific or multiple alert** names to filter the list.

  Figure 1-105: Alert Name



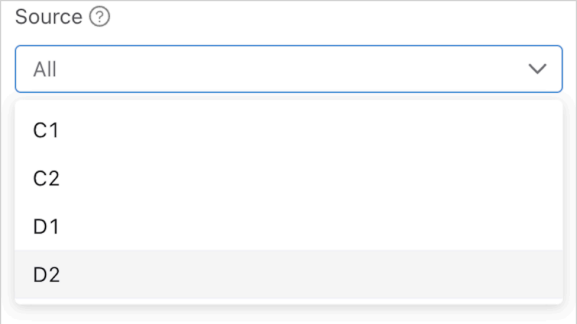
- **Source Type**: Use the drop-down menu to select a specific type of **source** (e.g., switch, policy, etc.).

  Figure 1-106: Source Type



- **Source**: If a **Source Type** is selected, the **Source** drop-down lists all the source names (e.g., switch names, policy names, etc.) used by the alert source as filters.

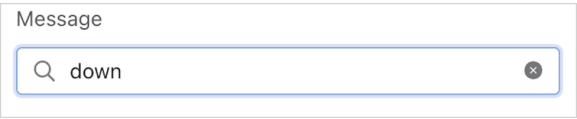
  Figure 1-107: Source



Note: The **Source** drop-down is enabled only when selecting one **Source Type**.

- **Message**: Enter keywords or phrases into the search text box to filter alerts based on their message content.

  Figure 1-108: Message



Reset Filters: In the **Alert Filters** section, select **Reset Filters** to clear all applied filters and view all alerts again.

Alert List

The **Alert List** section provides a comprehensive view of all active alerts based on the applied filters.

- **Columns:** The list includes the following columns:
  - **Name:** The name of the alert.
  - **Source:** The origin of the alert (e.g., policy name, switch name, etc.).
  - **Message:** A brief description of the alert.
- **Sorting:** Sort the list by **Name** or **Source** by selecting the respective column headers. An arrow indicates the current sort order (ascending or descending).
- **Expanding Alert Details:** For some alerts, an **expand** icon appears next to the alert name. Selecting the icon displays additional details about the alert.

Figure 1-109: LAG Member Missing

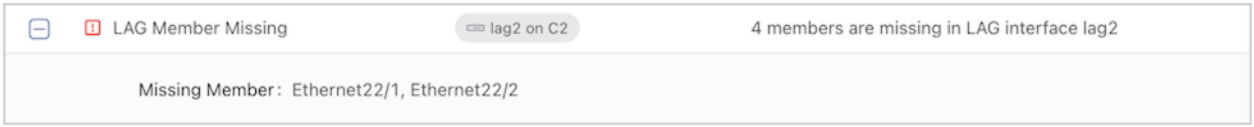
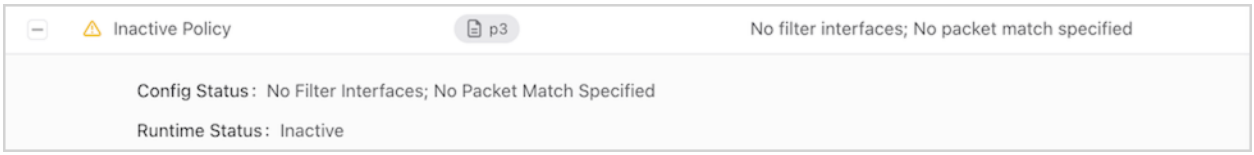


Figure 1-110: Inactive Policy



- **Export**: Select **Export** to download the current list of alerts to a file (CSV or JSON file), which can be useful for further analysis or record-keeping.

Important: New alerts have been introduced to enhance fabric health monitoring. DMF 8.8.0 now includes an **Inactive Policy** warning alert. This specific alert is not displayed when using the CLI **`show fabric warnings`** command.

1.3 Using the Command Line Interface

Before connecting to the DANZ Monitoring Fabric (DMF) Controller, ensure the DMF application is running. Log in to the DMF Controller using the local console or SSH to the address assigned to the DMF Controller during installation.



Note: Make all fabric switch configuration changes using the Controller CLI, which provides configuration options in the **config-switch** submode for each switch. Do not log in to the switch to make changes directly using the switch CLI.

Mode and sub-modes divide CLI commands, which restrict commands to the appropriate context. The main modes and their available commands are as follows:

- **login mode**: commands available immediately after logging in, with the broadest possible context.
- **enable mode**: commands that are available only after entering the enable command.
- **config mode**: commands that significantly affect system configuration and can only be entered after entering the **configure** command. The user can also access submodes from this mode.

Enter submodes from config mode to provision specific monitoring fabric objects. For example, the switch **switchname** command changes the CLI prompt to **(config-switch)#** and lets the user configure the switch identified by the switch name.

When the user logs in via SSH to the Controller, the CLI appears in login mode, where the default prompt is the system name followed by a greater than sign (>), as shown below:

```
controller-1>
```

To change the CLI to enable mode, enter the **enable** command. The default prompt for enable mode is the system name followed by a pound sign (#), as shown below:

```
controller-1> enable
controller-1#
```

To change to config mode, enter the `configure` command. The default prompt for config mode is the system name followed by `(config)#`, as shown below:

```
controller-1> config
controller-1(config)#
```

To change to a submode, enter the command from config mode, followed by any object identifier required, as in the following example:

```
controller-1(config)# switch filter-switch-1
controller-1(config-switch)# interface ethernet54
controller-1(config-switch-if)#
```

To return to enable mode, type **end**, as shown below:

```
controller-1(config)# end
controller-1#
```

To view the path to the current CLI prompt, enter the **show this** command from any nested submode, as in the following example:

```
controller-1(config-switch-if)# show this
! switch
switch filter-switch-1
interface ethernet54
```

To view details about the configuration, enter the **show this details** command, as in the following example:

```
controller-1(config-switch-if)# show this details
! switch
switch filter-switch-1
!
interface ethernet54
no force-link-up
no optics-always-enabled
no shutdown
```

To view a list of available commands in the current or submode, enter the **help** command.

```
controller-1> help
For help on specific commands: help <command>
Commands:
%<n>                    Move job to foreground
debug
echo                    Print remaining arguments
enable                  Enter enable mode
exit                    Exit submode
help                    Show help
history                 Show commands recently executed
logout                  Logout
no                      Prefix existing commands to delete item
ping                    Send echo messages
ping6                   Send echo messages
profile                 Configure user profile
reauth                  Reauthenticate
set                     Manage CLI sessions settings
show
support                 Generate diagnostic data bundle for technical support
terminal                Manage CLI sessions settings
topic                   Show documentation on topic
upload                  Upload diagnostic data bundle for technical support
watch                   Show output of other commands
whoami                  Identify the current authenticated account
workflow                Show workflow documentation
controller-1>
```

To view detailed online help for the command, enter the **help** command followed by the command.

```
controller-1> help support
Support Command:         Generate diagnostic data bundle for technical support
Support Command Syntax:  no support skip-switches skip-cluster skip-service-nodes
                         skip-recorder-nodes sequential support [[skip-switches]
                         [skip-cluster] [skip-service-nodes]
                         [skip-recorder-nodes] [sequential]]
Next Keyword Descriptions:
sequential:              Use sequential (non-parallel) fallback collection mode, which will be
                         slower but use fewer resources.
skip-cluster:            Skip cluster information from the collection.
skip-recorder-nodes:     Skip recorder nodes information from the collection.
skip-service-nodes:      Skip service nodes information from the collection.
skip-switches:           Skip switches information from the collection.
Support Command:         Generate diagnostic data bundle for technical support
Support Command Syntax:  no support skip-switches skip-cluster skip-service-nodes
                         skip-recorder-nodes sequential support [[skip-switches]
                         [skip-cluster] [skip-service-nodes] [skip-recorder-nodes] [sequential]]
Next Keyword Descriptions:
sequential:              Use sequential (non-parallel) fallback collection mode, which will be
                         slower but use fewer resources.
skip-cluster:            Skip cluster information from the collection.
skip-recorder-nodes:     Skip recorder nodes information from the collection.
skip-service-nodes:      Skip service nodes information from the collection.
skip-switches:           Skip switches information from the collection.
controller-1>
```

To display the options available for a command or keyword, enter the command or keyword followed by a question mark (?).

```
controller-1> support ?
<cr>
sequential               Use sequential (non-parallel) fallback collection mode, which will be
                         slower but use fewer resources.
skip-cluster             Skip cluster information from the collection.
skip-recorder-nodes      Skip recorder nodes information from the collection.
skip-service-nodes       Skip service nodes information from the collection.
skip-switches            Skip switches information from the collection.
controller-1>
```

To view any command's permitted values or keywords, enter the command followed by a space and press the <Tab> key. The command completion feature displays a concise list of permitted values, as in the following example:

```
controller-1> support <TAB>
<cr> sequential skip-cluster
skip-recorder-nodes skip-service-nodes skip-switches
controller-1>
```

For information about managing administrative access to the DMF Controller, refer to the ***DANZ Monitoring Fabric Deployment Guide***.

Chapter 2

Managing DMF Switches and Interfaces

This chapter describes the basic configuration required to deploy and manage DANZ Monitoring Fabric (DMF) switches and interfaces.

2.1 DMF Switches Dashboard

DANZ Monitoring Fabric (DMF) 8.7 introduces an updated Switches dashboard. A header and tabulated layout enable observation of different aspects of the installed switches and provisioning new switches, all within the same dashboard.

Using the DMF Switches Dashboard

Overview

Inventory

Configuration

Properties

IP Address Allocation

TCAM Utilization

Firmware

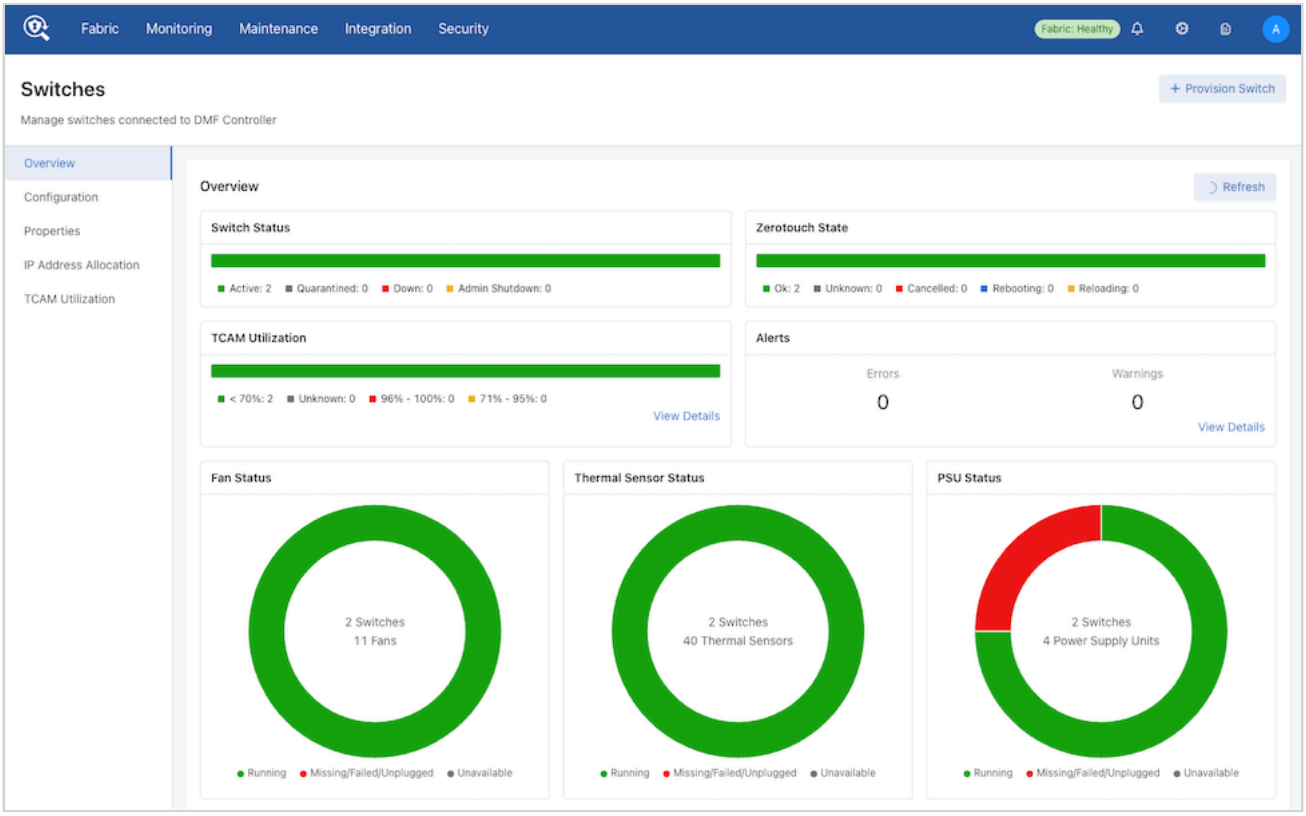
Revert Switches Dashboard

2.1.1 Overview

Select **Fabric** > **Switches** > **Overview**.

The **Overview** dashboard comprises two separate sections: **Overview** and **Inventory**.

Figure 2-1: DMF Switches Overview



The **Overview** section has seven charts.

- **Switch Status**: Displays the summary of switch connection status.
- **Zerotouch State**: Displays the summary of Zerotouch state of all switches.
- **TCAM Utilization**: Displays the TCAM usage of all switches. The **View Details** link navigates to the **TCAM Utilization and Capacity** view.
- **Alerts**: Displays the count of relevant switch alerts and warnings. The **View Details** link navigates to the **Alerts** page.
- **Fan Status**: Displays the switch count, and fan count; a donut chart shows the percentage of fans with an unhealthy status.
- **Thermal Sensor Status**: Displays the switches count, thermal sensors count and a donut chart showing the percentage of sensors with an unhealthy status.
- **PSU (Power Supply Unit) Status**: displays the switches count, PSU count, and a donut chart showing the percentage of PSUs with unhealthy status.

Hovering over a bar chart or an arc renders a tooltip explaining the current arc and count.

Figure 2-2: Hover Details Bar Chart

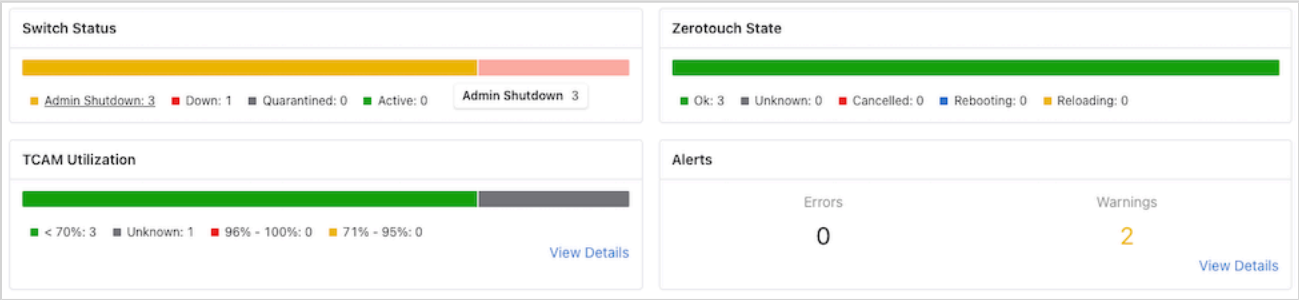
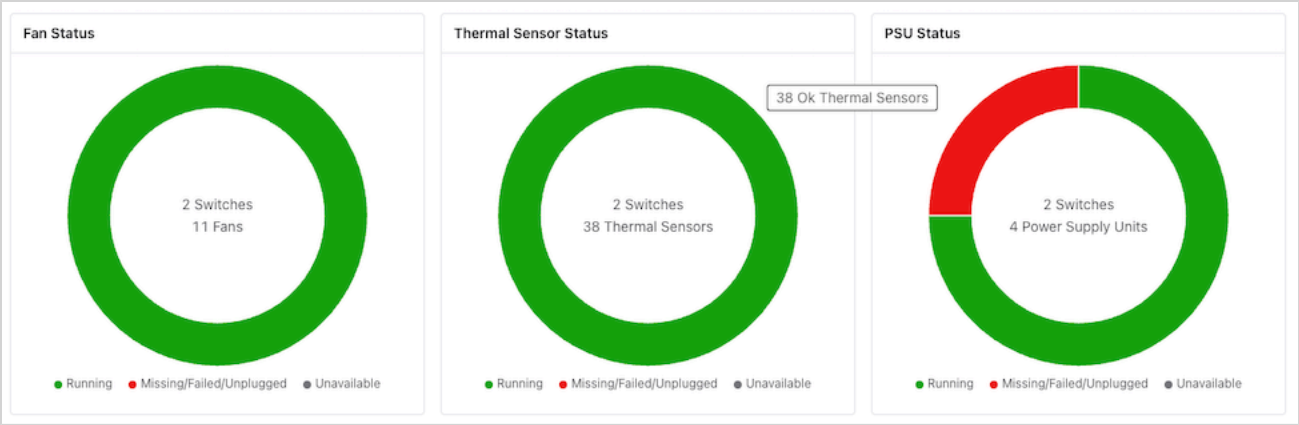
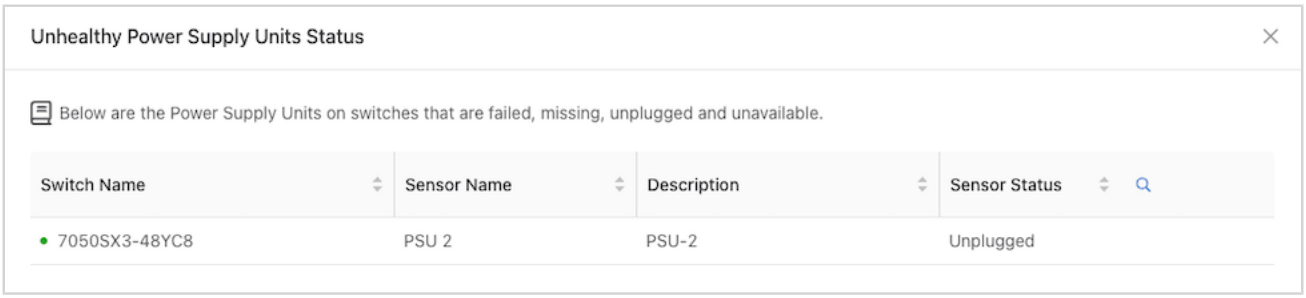


Figure 2-3: Hover Details Arc



Selecting an unhealthy arc in **Fan Status**, **Thermal Sensor**, or **PSU**, opens a status window listing the items with an abnormal status or if the status is unavailable.

Figure 2-4: Status



2.1.2 Inventory

The **Inventory** section displays all inventory records configured in the switches. Table functionality includes exporting, filtering, sorting, hiding, or showing columns.

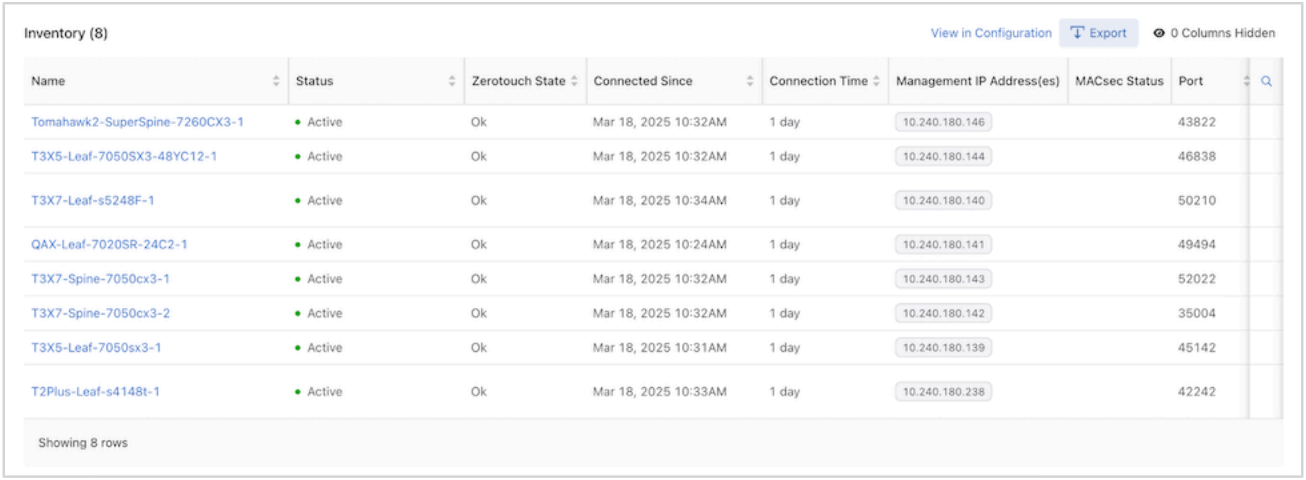
Selecting a link in the **Name** column opens a **Switch Detail** window.

Figure 2-5: Switch Detail



Select **View in Configuration** to view the switches in the **Configuration** mode.

Figure 2-6: View in Configuration



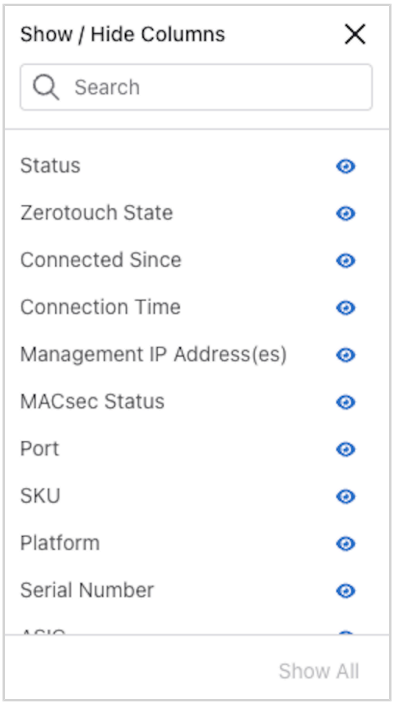
Columns include:

Name	Status	Zerotouch State
Connected Since	Connection Time	Management IP Address(es)
MACsec Status	Port	SKU
Platform	Serial Number	ASIC
Software Description	Pipeline Mode	

Show / Hide Columns determine the information displayed in the dashboard and are user-selectable. Selections persist until changed.

Note: The **Name** column cannot be hidden.

Figure 2-7: Show / Hide Columns



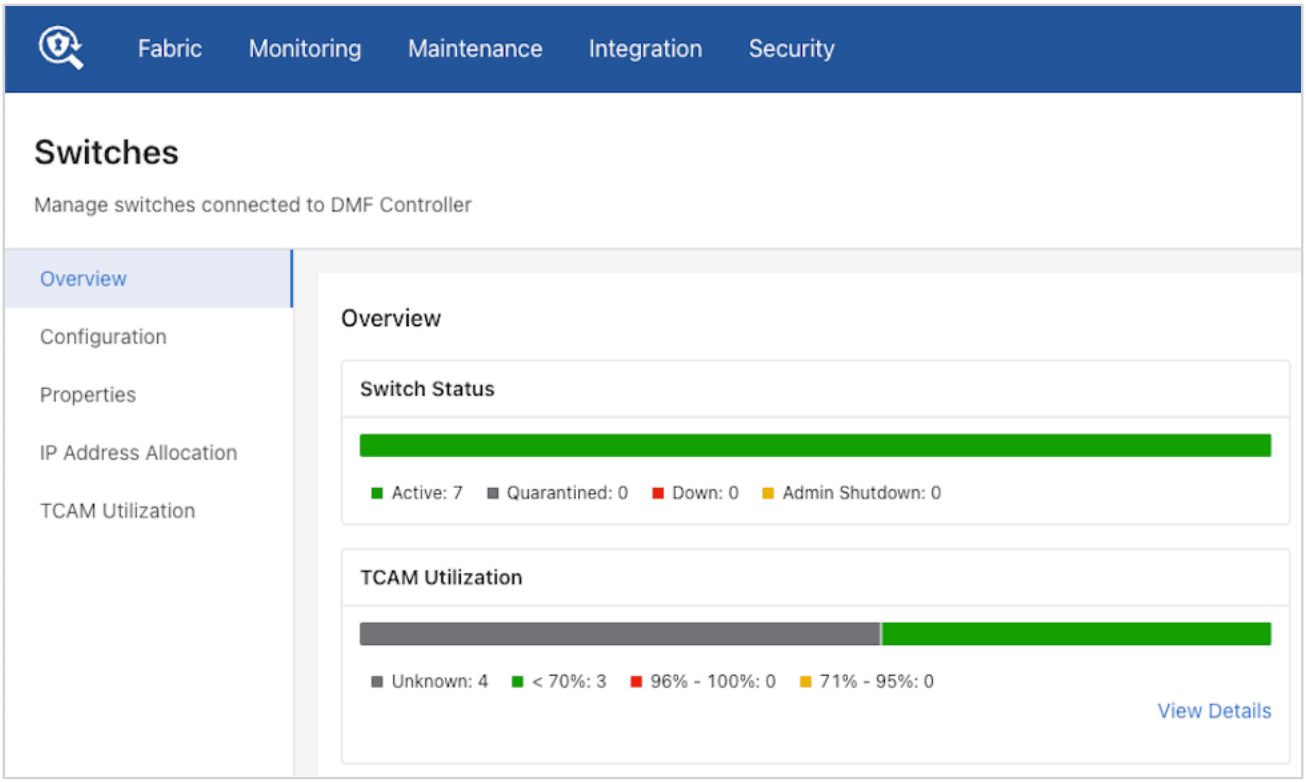
Search: Filter rows with matching text.

2.1.2.1 Optic Cable Transceiver Detail Information

The **Switch** detail page in the DMF GUI has an **Inventory** tab displaying information about optics, cables, and transceivers.

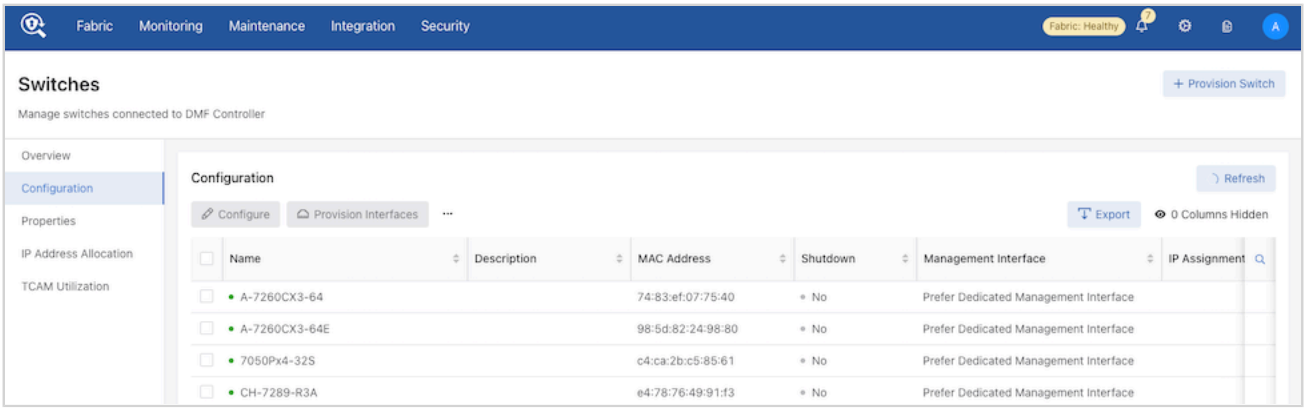
Select **Fabric** > **Switches** > **Configuration** to display a list of fabric switches.

Figure 2-8: Switches Overview



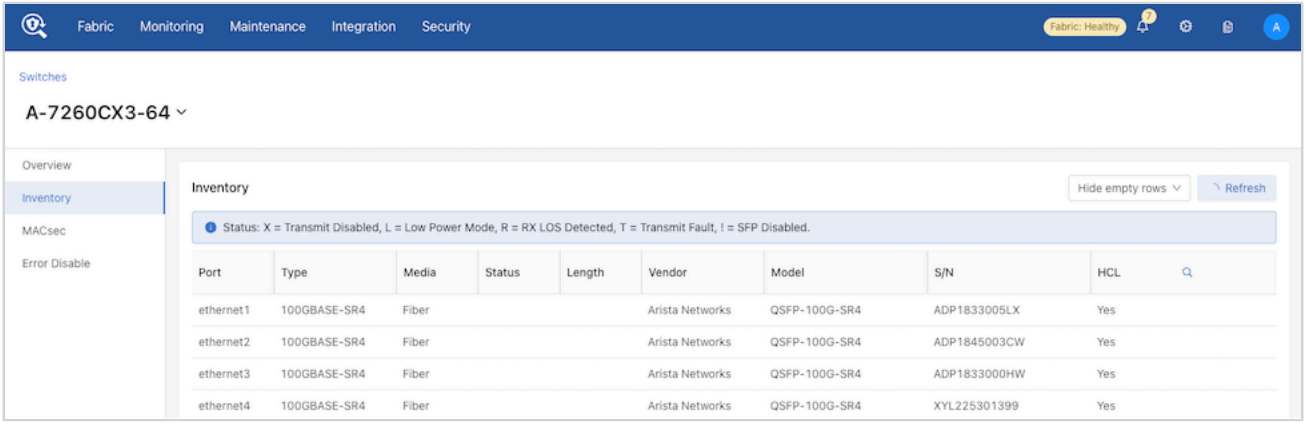
Choose a switch from the **Configuration** list.

Figure 2-9: List of Switches

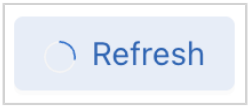


Select **Inventory** to view the inventory details.

Figure 2-10: Inventory



Refresh Button



The data in the **Inventory** tab refreshes automatically every 60 seconds. However, you can manually refresh the data by using **Refresh** as needed.

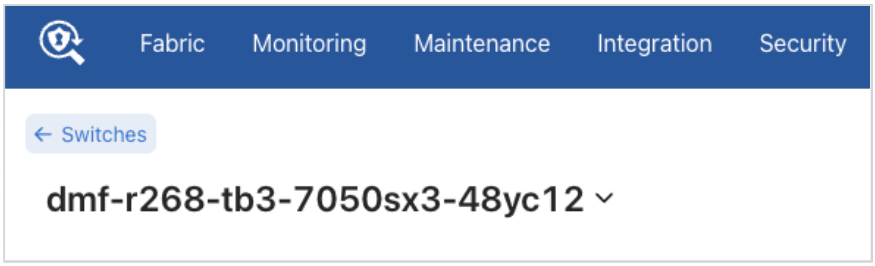
The inventory data for SwitchLight (SWL) and EOS switches display differently.

Header

The header contains two interactive components.

- A **Switches** back button that returns user to the switches page or configuration table.
- A **Select** drop-down to conveniently view the inventory of other switches and allows text and search.
- **Back Button** & **Select Drop-down**.

  Figure 2-11: Back Button & Select Drop-down



SwitchLight (SWL) Switch

DMF displays SwitchLight switches inventory information as a table. **Search** functionality allows quick entry lookup.

There is an option to hide and show the table with empty rows.

Figure 2-12: Example - Table with Empty Rows Hidden

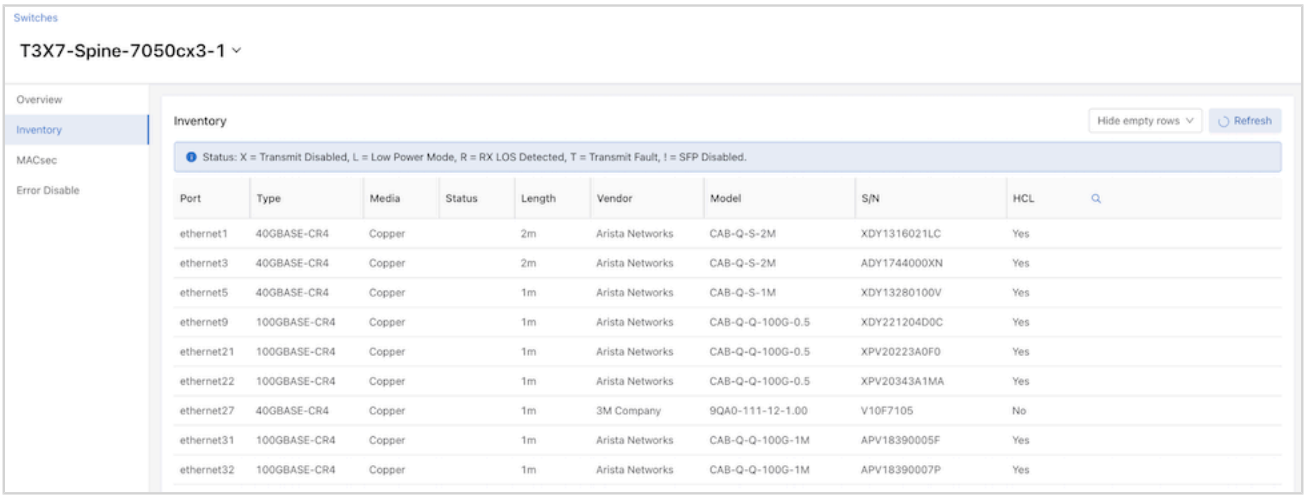
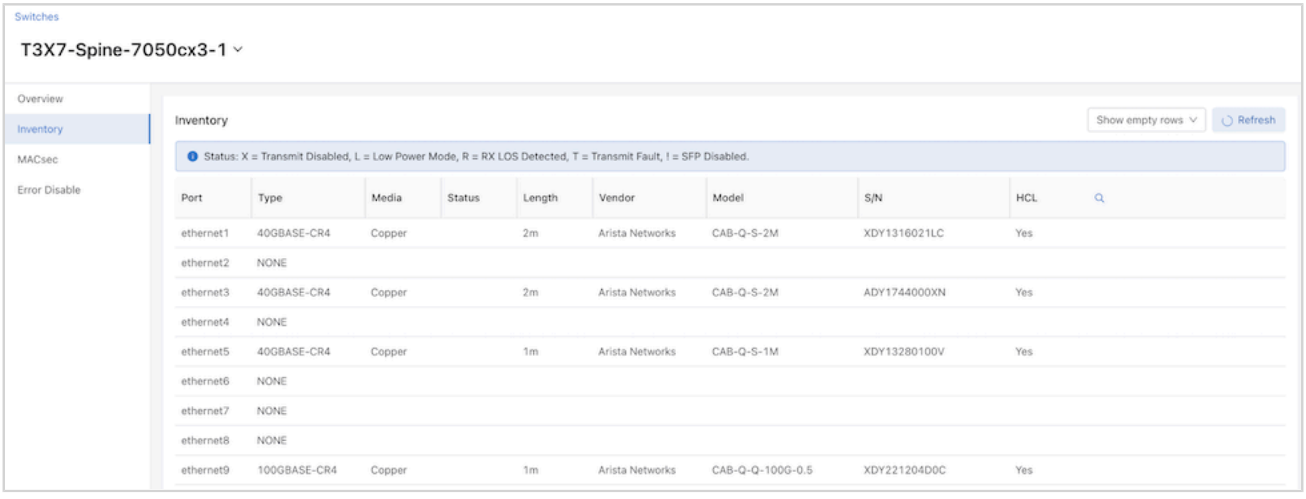


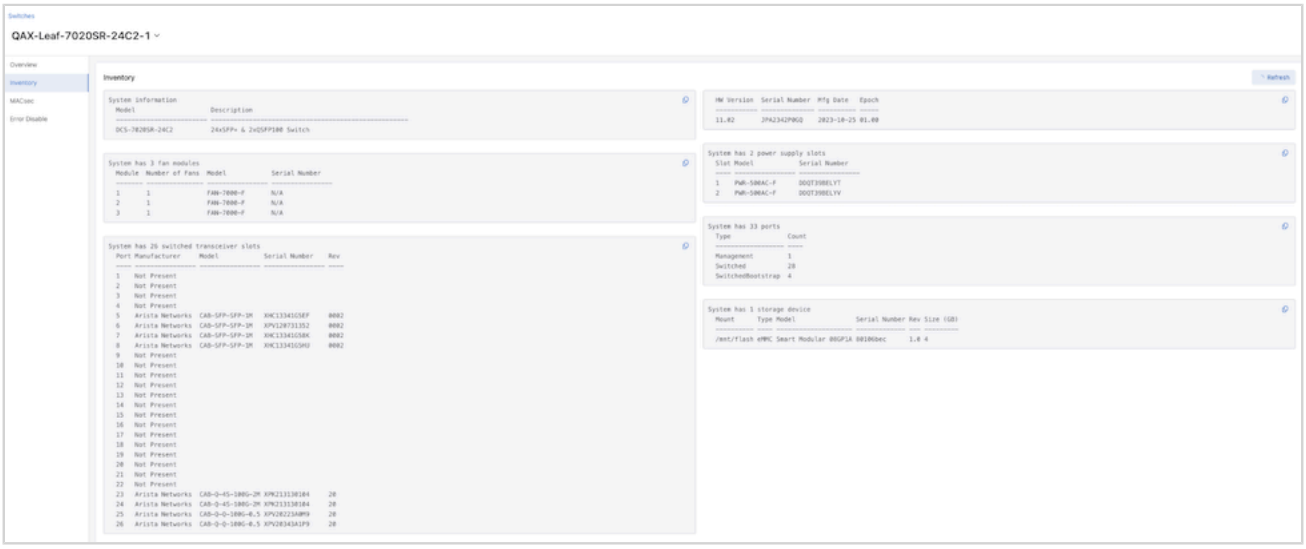
Figure 2-13: Example - Table with Empty Rows Shown



EOS Switch

DMF displays the inventory information for EOS switches in block format. Each block displays a different aspect of the inventory and contains a **copy** icon enabling quick data copying.

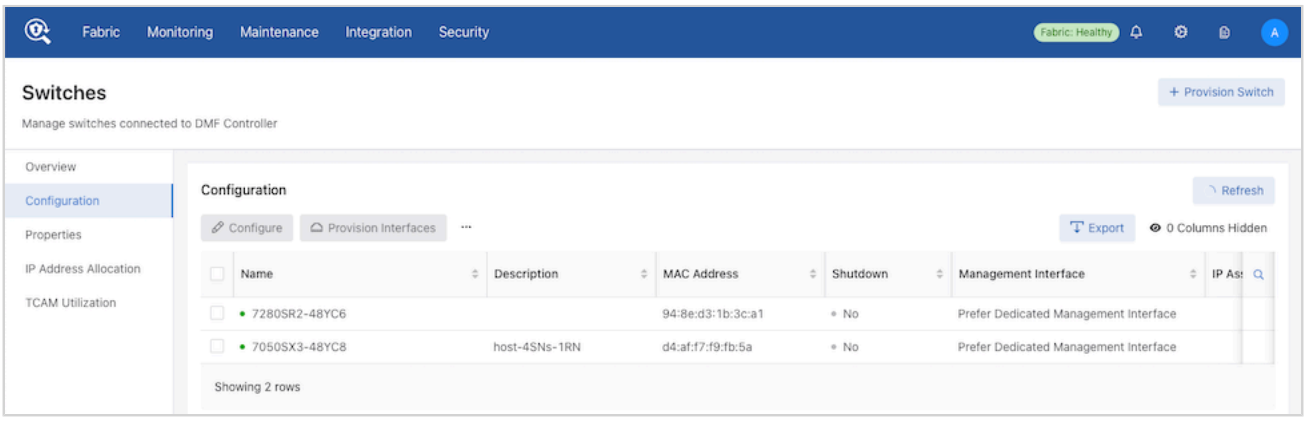
Figure 2-14: EOS Switch Details



2.1.3 Configuration

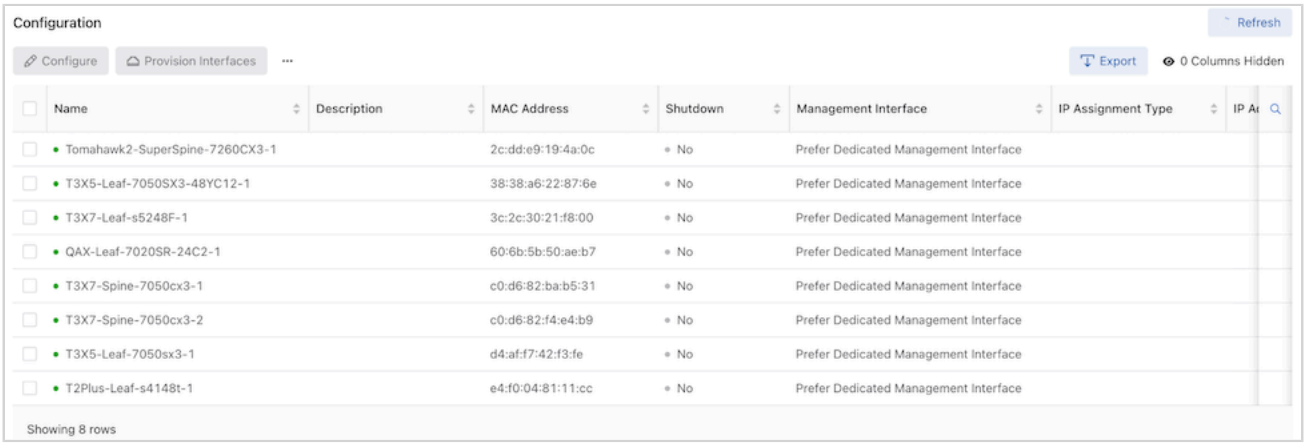
Select **Fabric** > **Switches** > **Configuration**.

Figure 2-15: Switches Configuration



Configuration Details: A sortable display of all configurations of the switches currently installed in DMF.

Figure 2-16: Configuration Details



Columns include:

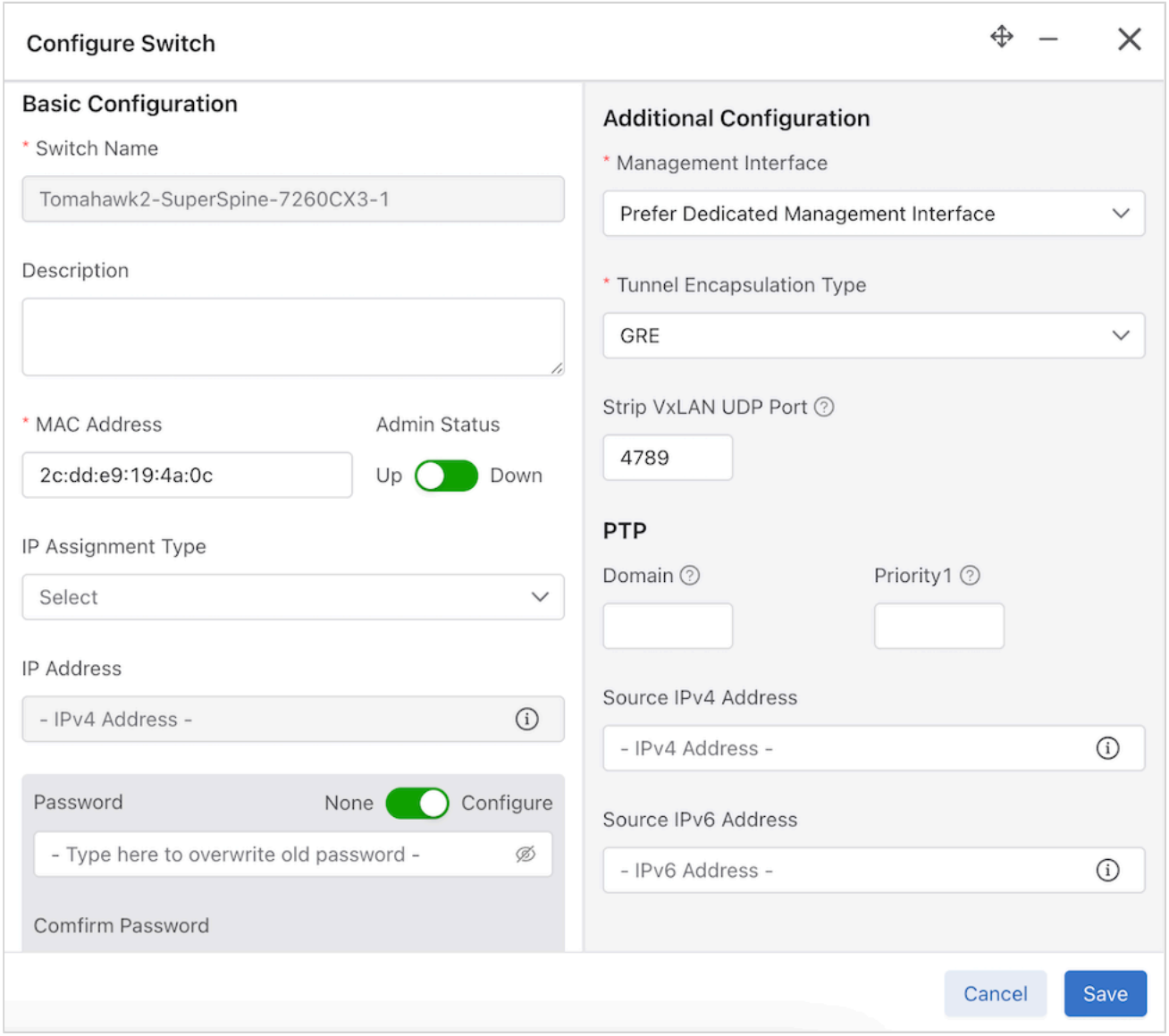
Name	Description
MAC Address	Shutdown
Management Interface	IP Assignment Type
IP Address	Tunnel Encapsulation Type
Strip VXLAN UDP Port	Password
PTP Domain	PTP Priority1
PTP IPv4 Address	PTP IPv6 Address

The **Configuration** dashboard comprises the following components and controls:

- **Refresh**: By default, the system refreshes data every 60 seconds. Hovering over **Refresh** displays the time remaining until the next update. When required, select **Refresh** to update the data immediately.
- **Configure**: Opens the **Configuration Switch** window with pre-populated switch information. Edit the configuration as required and select **Save**.

Configure is disabled unless one switch is selected.

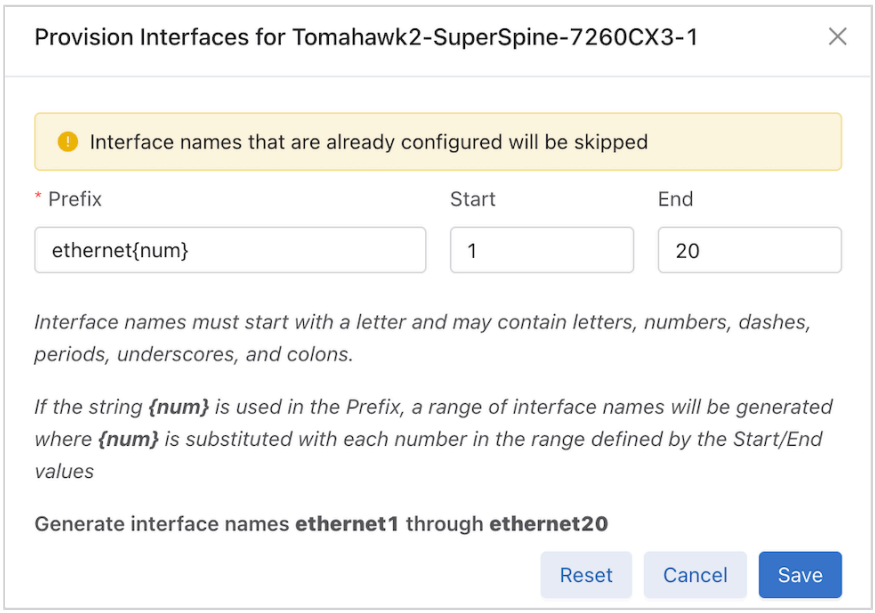
Figure 2-17: Configure Switch



- **Provision Interfaces**: Provides a fast and convenient way to populate a large number of interfaces. Prefix works similarly to template string formatting, where the **`{num}`** is replaced by a number between start and end, if present.

Provision Interfaces is disabled unless one switch is selected.

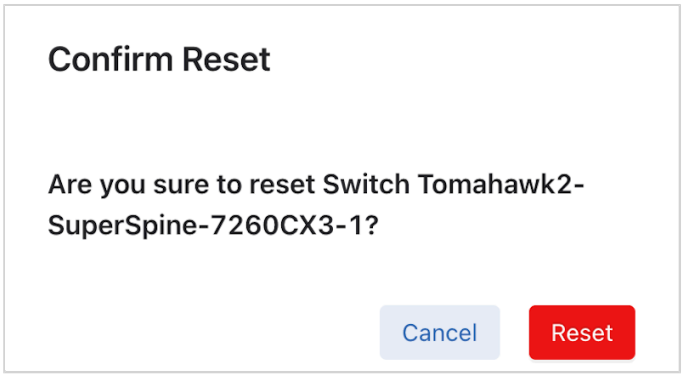
Figure 2-18: Provision Interfaces



- **Reset**: Disconnect and reset the connection to the switch.

  Reset is disabled unless one switch is selected.

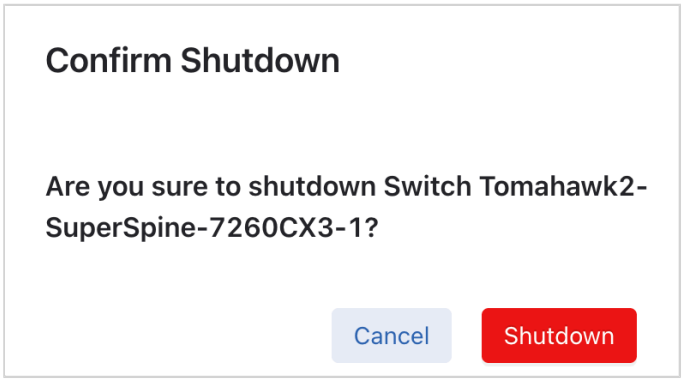
Figure 2-19: Confirm Reset



- **Shutdown**: Sets shutdown property of the switch config to true, thus shutting down the switch.

  Shutdown is disabled unless one switch is selected.

  Figure 2-20: Confirm Shutdown



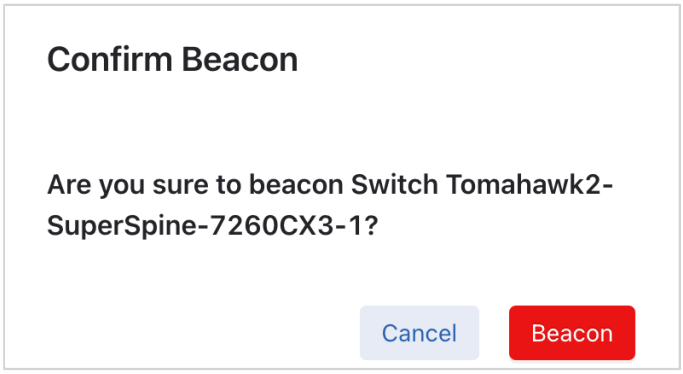
- **Reboot**: Reboot the zerotouch device.

  Reboot is disabled unless one switch is selected and the selected switch is a ZTN device. For a Chassis, select the **Module Type** and the **Slot** to reboot.

- **Beacon**: Beacons the LED light of the zerotouch device for a certain period.

  Beacon is disabled unless one switch is selected and the selected switch is a ZTN device.

  Figure 2-21: Confirm Beacon

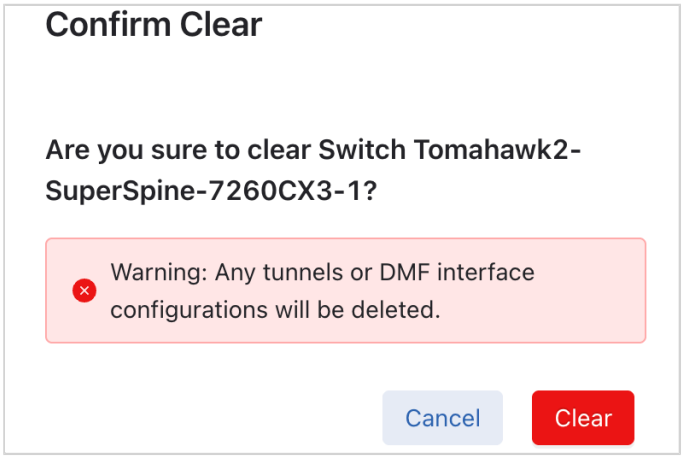


  For a Chassis, select the **Module Type**, **Slot**, and **Option** to beacon.

- **Clear Config**: Deletes the current switch configuration.

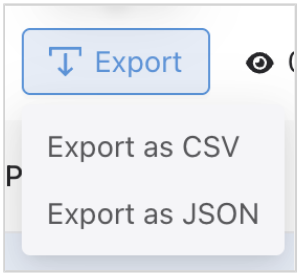
  Clear is disabled unless one switch is selected.

  Figure 2-22: Config Clear



- **Export**: Export the data in CSV or JSON format.

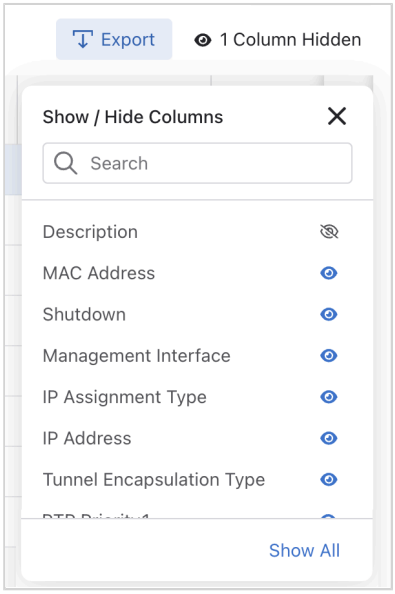
  Figure 2-23: Export Data



- **Show / Hide Columns** determine the information displayed in the dashboard and are user-selectable. Selections persist until changed.

> **Note:** The **Name** column cannot be hidden.

Figure 2-24: Show / Hide Columns



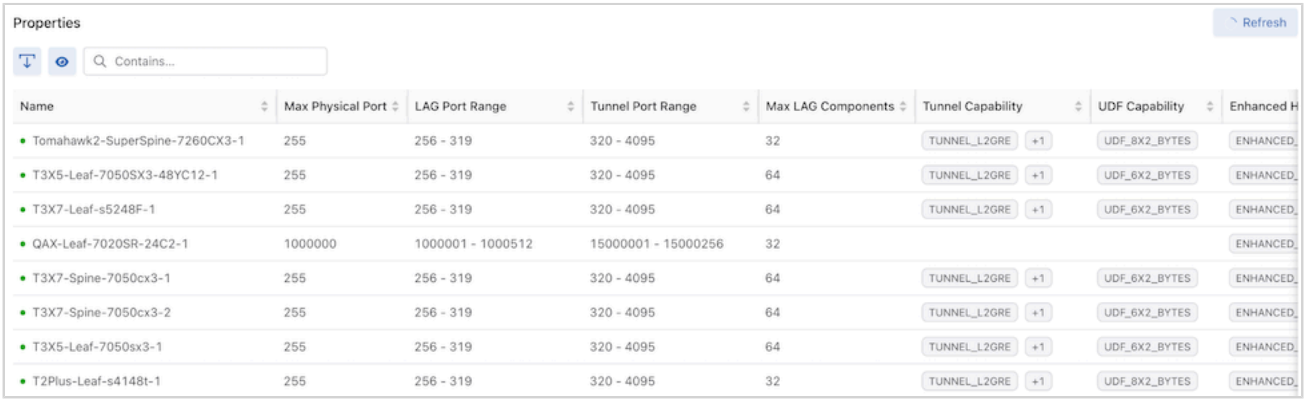
- **Search**: Filter rows with matching text.

2.1.4 Properties

Select **Fabric** > **Switches** > **Properties**.

Properties Dashboard: A sortable display of all switch properties currently installed in DMF.

Figure 2-25: Properties Dashboard



Columns include:

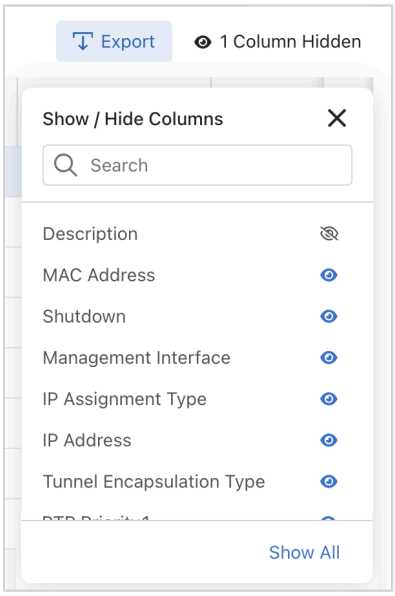
Name	Max Physical Port
LAG Port Range	Tunnel Port Range
Max LAG Components	Tunnel Capability
UDF Capability	Enhanced Hash Capability
Rate Limit Range	Max Multicast Replication Groups
Max Multicast Replications	Supports CPU Trap Table
PTP Timestamp Capability	Truncate Range
Strip Header Capability	MACsec Capability
Port Count	Egress Strip VLAN Capability

The **Properties** dashboard comprises the following components and buttons:

- **Refresh**: By default, the system refreshes data every 60 seconds. Hovering over **Refresh** displays the time remaining until the next update. When required, select **Refresh** to update the data immediately.
- **Actions:**
  - **Export**: Export data in CSV or JSON format.

- **Show / Hide Columns** determine the information displayed in the dashboard and are user-selectable. Selections persist until changed.

  Figure 2-26: Show / Hide Columns



Note: The **Name** column cannot be hidden.

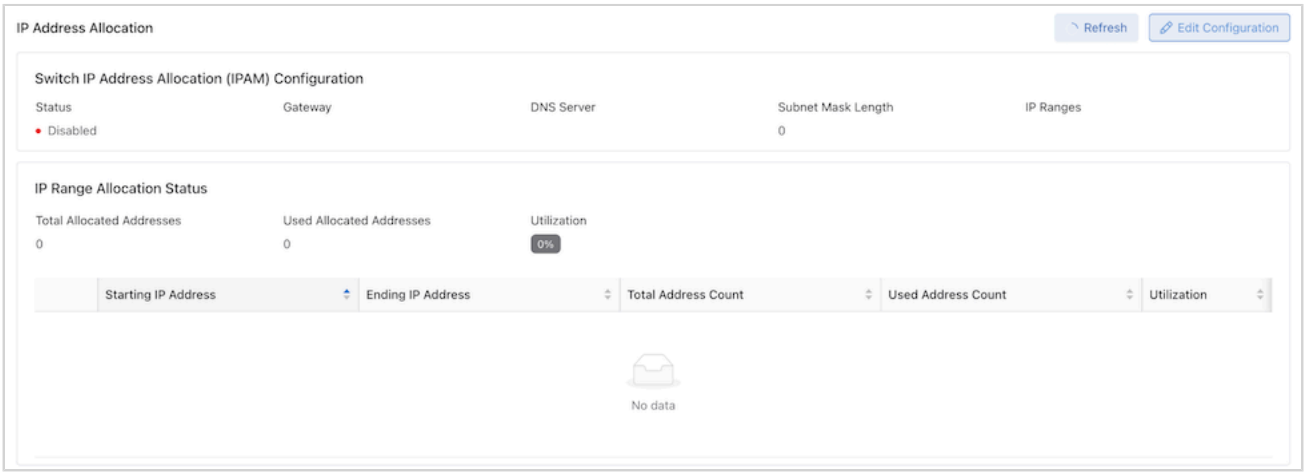
- **Search**: Filter rows with matching text.

2.1.5 IP Address Allocation

Select **Fabric** > **Switches** > **IP Address Allocation**.

The **IP Address Allocation** dashboard displays the current switches' **IP Address Allocation Configuration** and **IP Range Allocation Status**.

Figure 2-27: IP Address Allocation



Columns include:

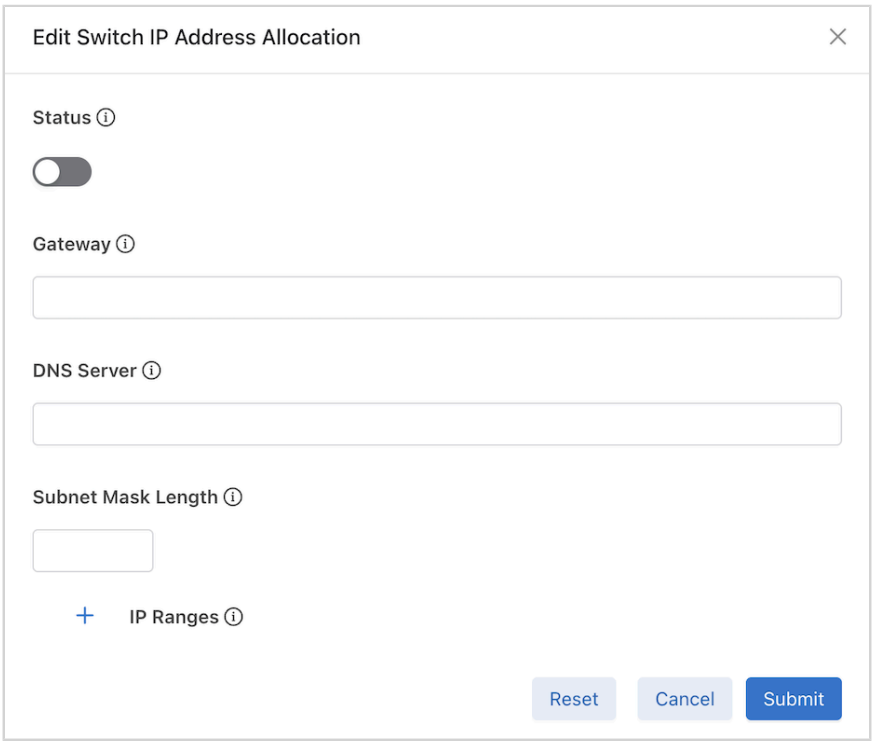
Starting IP Address	Ending IP Address	Total Address Count	Used Address Count	Utilization

Select **Edit Configuration** to configure or modify **IP Address Allocation**.

Use **Status** to enable or disable the feature.

Enter the required information and select **Submit**. Optionally, select **Reset** to clear the information and start again.

Figure 2-28: Edit Switch IP Address Allocation

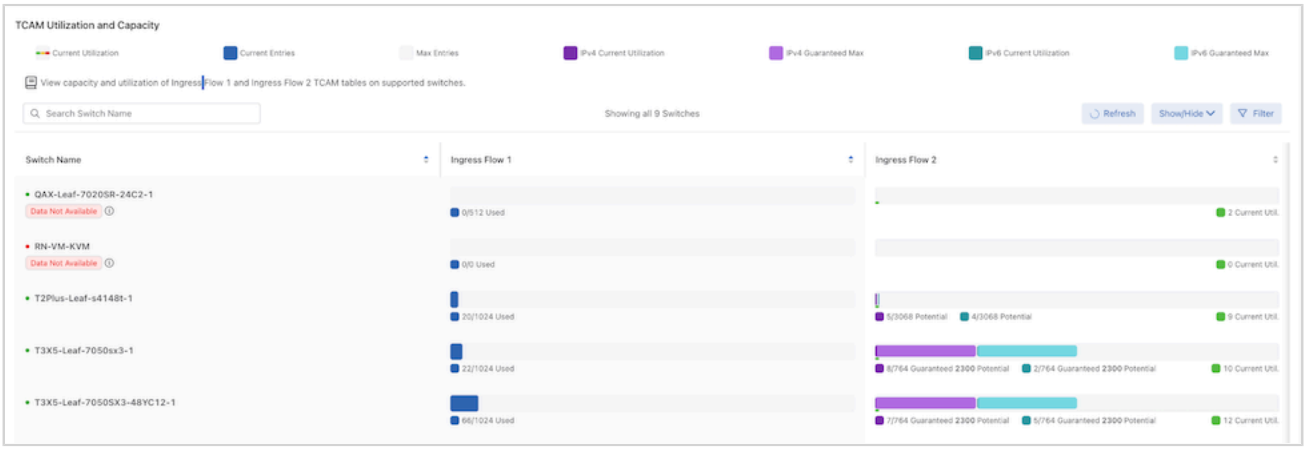


2.1.6 TCAM Utilization

Select **Fabric** > **Switches** > **TCAM Utilization**.

The **TCAM Utilization and Capacity** dashboard displays the TCAM utilization and capacity of the current switches. View capacity and utilization of **Ingress Flow 1** and **Ingress Flow 2** TCAM tables on supported switches.

Figure 2-29: TCAM Utilization and Capacity



Columns include:

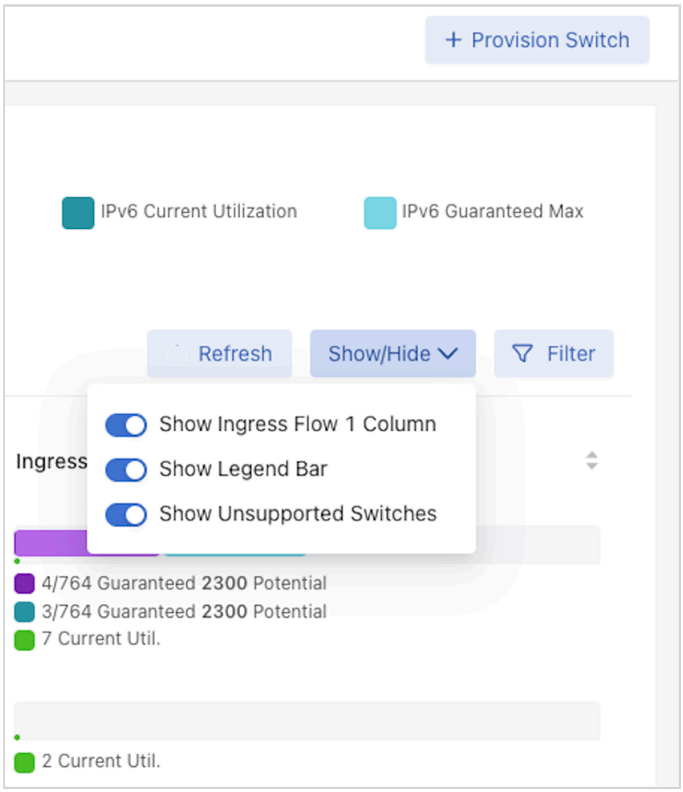
Switch Name	Ingress Flow 1	Ingress Flow 2

The **TCAM Utilization and Capacity** dashboard comprises the following components and buttons:

- **Refresh**: By default, the system refreshes data every 60 seconds. Hovering over **Refresh** displays the time remaining until the next update. When required, select **Refresh** to update the data immediately.

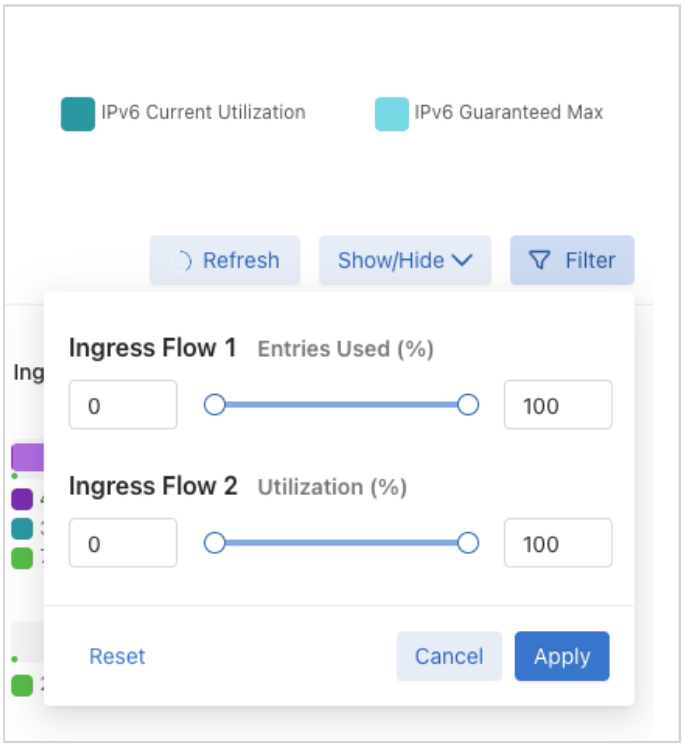
- **Show / Hide Columns** determine the information displayed in the dashboard and are user-selectable. Selections persist until changed.

  Figure 2-30: Show / Hide Columns



- **Filter**: Filter on **Ingress Flow 1** and **Ingress Flow 2** by entering a value or moving the sliders and select **Apply**.

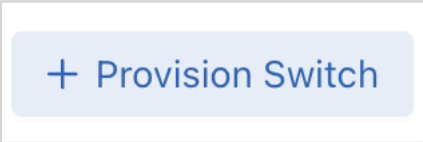
Figure 2-31: Filter Ingress Flow



2.1.7 Provision Switch

Select **Fabric** > **Switches** > **Provision Switch**.

Figure 2-32: Provision Switch Control

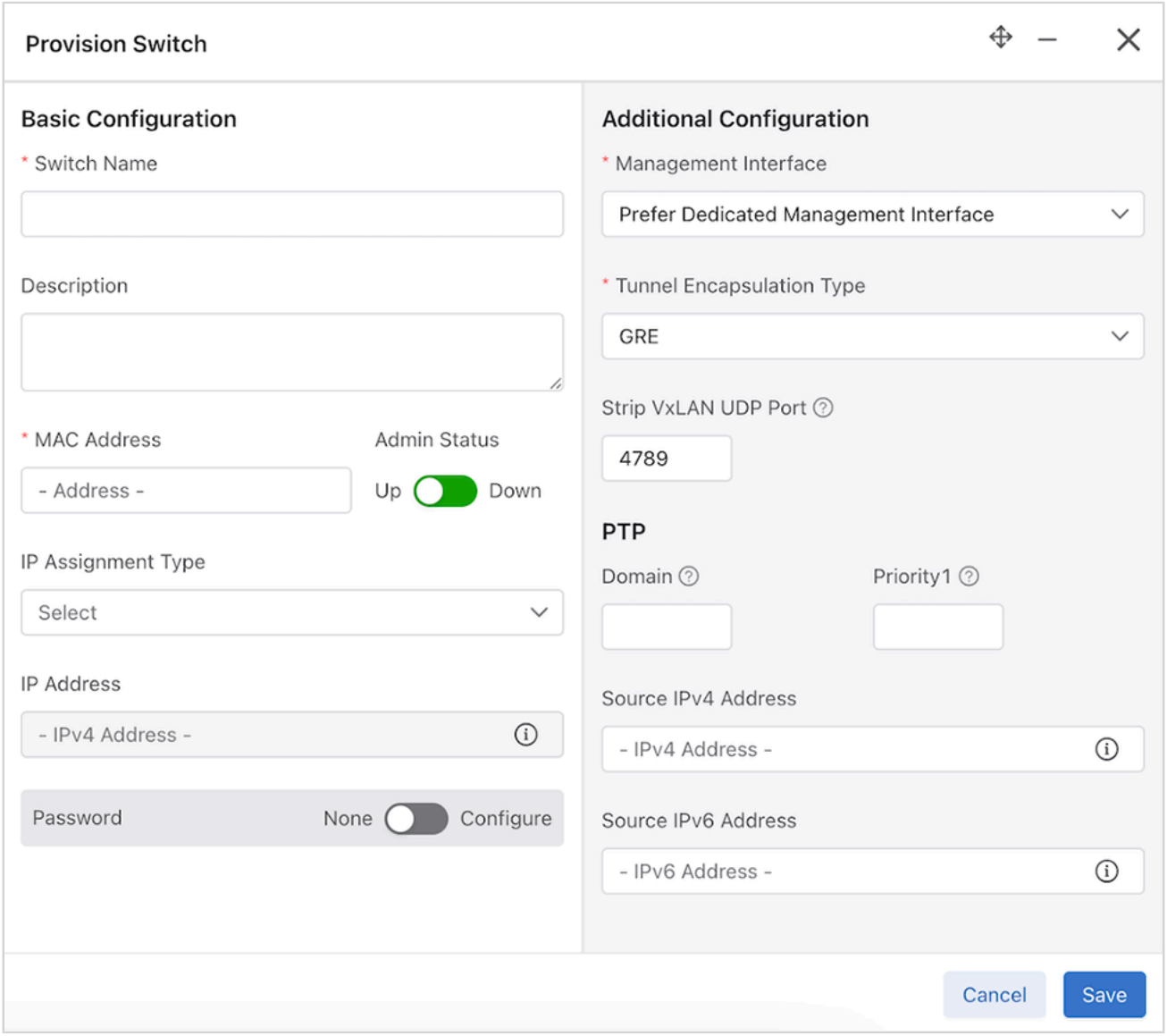


Provision Switch opens a window to provision a single switch. The required fields are:

- **Switch Name**
- **MAC Address**
- **Management Interface**
- **Tunnel Encapsulation Type**

The remaining fields are optional.

Figure 2-33: Provision Switch



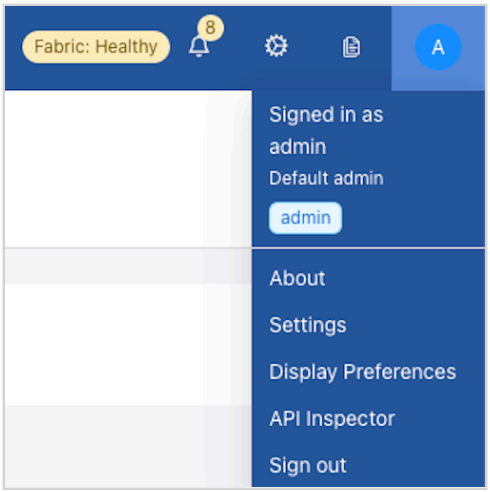
2.1.8 Firmware

The **Firmware** dashboard is hidden by default (Dell switches are not supported in the DMF 8.7.0 release).

If required, enable the **Firmware** dashboard using Switch Firmware on the **Settings** page.

To access the page, select the **Admin** icon followed by **Settings**.

Figure 2-34: Admin Settings

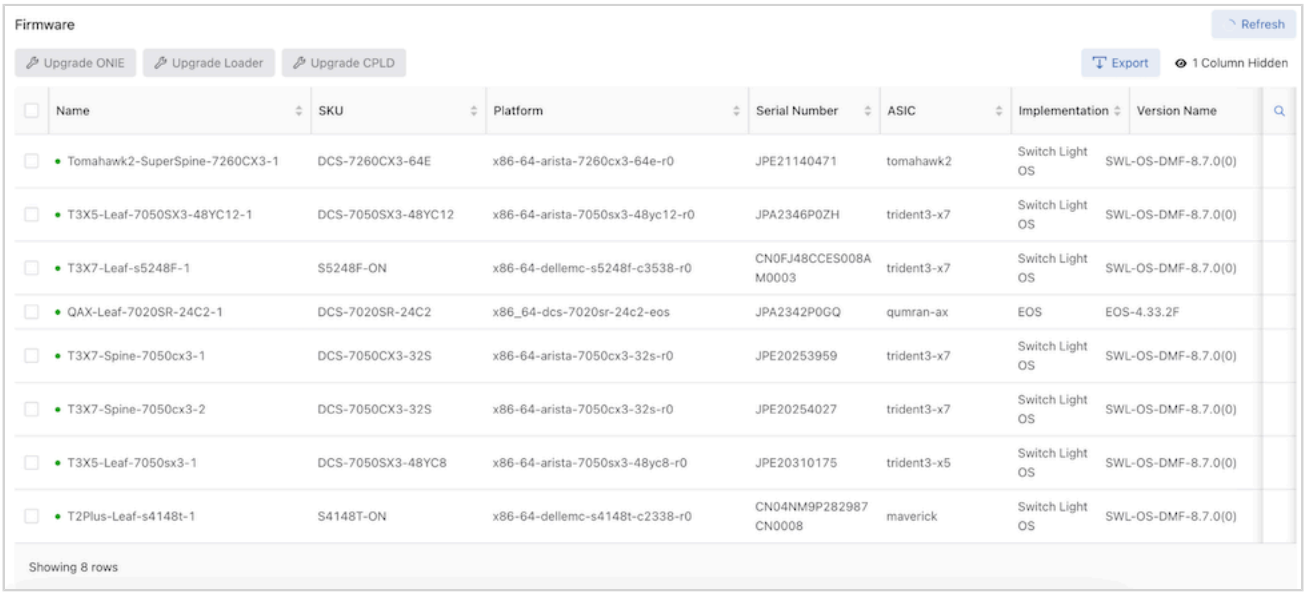


Firmware

Select **Fabric** > **Switches** > **Firmware**.

Firmware Dashboard: A sortable display of all switch firmware properties currently installed in DMF.

Figure 2-35: Firmware Dashboard



Columns include:

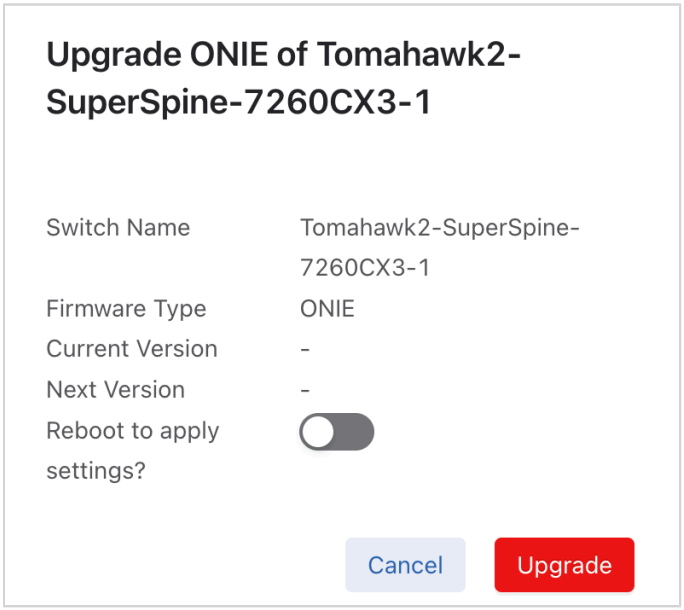
Name	Manufacturer
SKU	Platform
Serial Number	ASIC
Implementation	Version Name
Loader Version	Next Loader Version
CPLD	Next CPLD
ONIE	Next ONIE

The Firmware dashboard comprises the following components:

- **Refresh**: By default, the system refreshes data every 60 seconds. Hovering over **Refresh** displays the time remaining until the next update. When required, select **Refresh** to update the data immediately.
- **Actions:**
  - **Upgrade ONIE**: Upgrade the ONIE settings of the current switch from the **Current Version** to the **Next Version**.

    Upgrade is disabled unless one switch is selected.

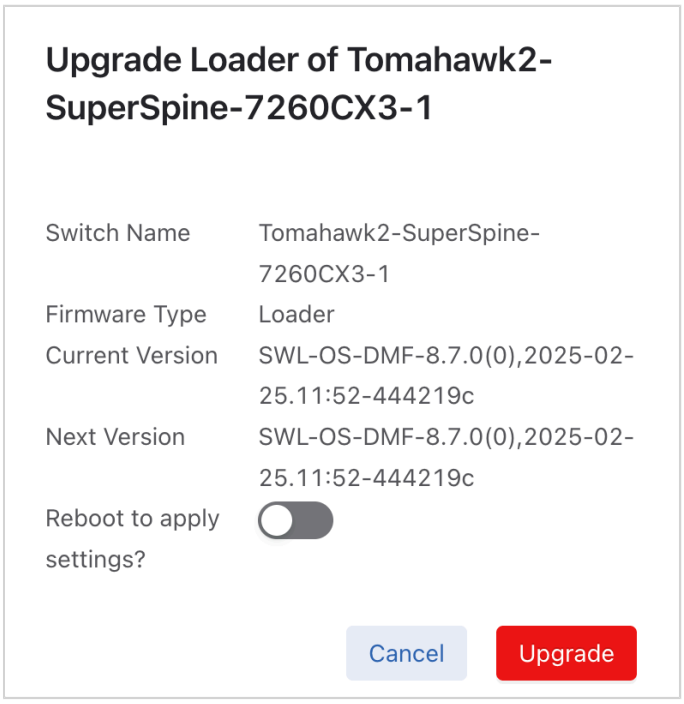
    Figure 2-36: Upgrade ONIE

  - **Upgrade Loader**: Upgrade the Loader settings of the current switch from the **Current Version** to the **Next Version**.

Upgrade is disabled unless one switch is selected.

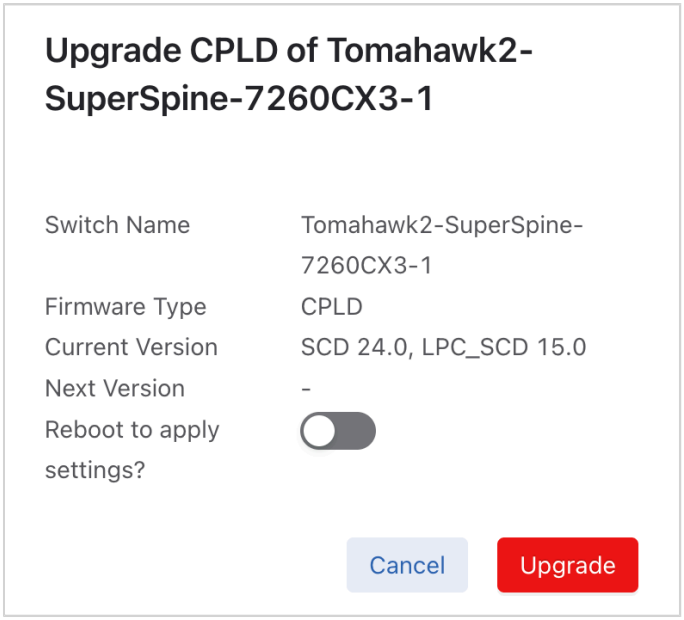
Figure 2-37: Upgrade Loader



- **Upgrade CPLD**: Upgrade the CPLD settings of the current switch from the **Current Version** to the **Next Version**.

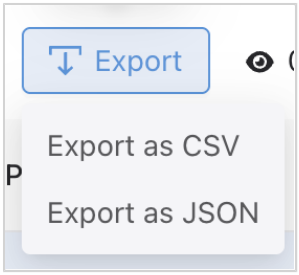
  Upgrade is disabled unless one switch is selected.

Figure 2-38: Upgrade CPLD



- **Export**: Export data in CSV or JSON format.

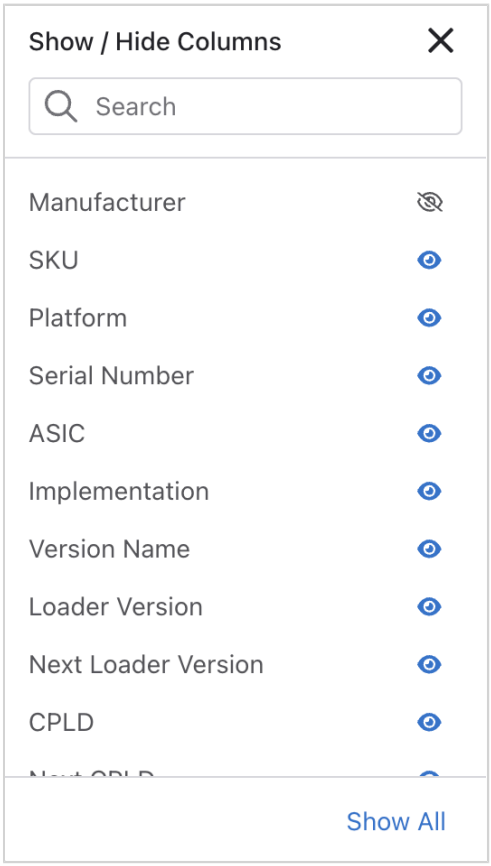
  Figure 2-39: Export Data



- **Show / Hide Columns** determine the information displayed in the dashboard and are user-selectable. Selections persist until changed.

> **Note:** The **Name** column cannot be hidden.

  Figure 2-40: Show / Hide Columns



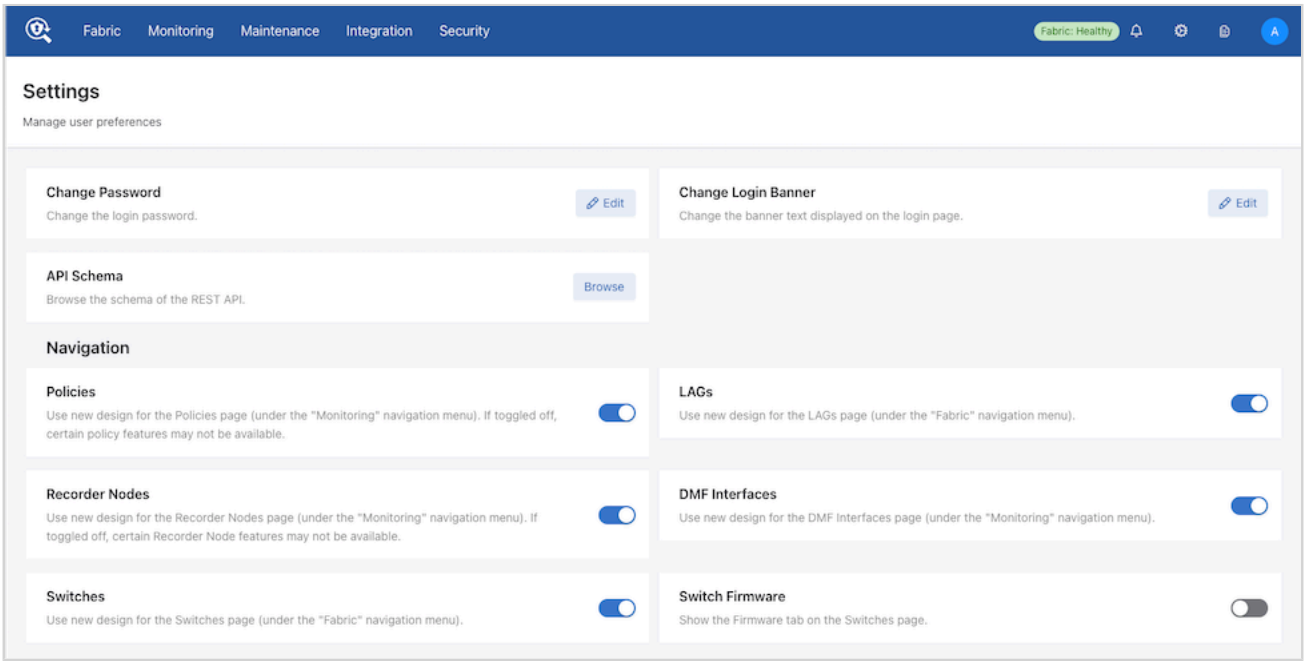
- **Search**: Filter rows with matching text.

2.1.9 Revert to the Legacy Dashboard

You can choose to remain on the legacy dashboard by toggling the **Switches** setting in the **Settings** dashboard.

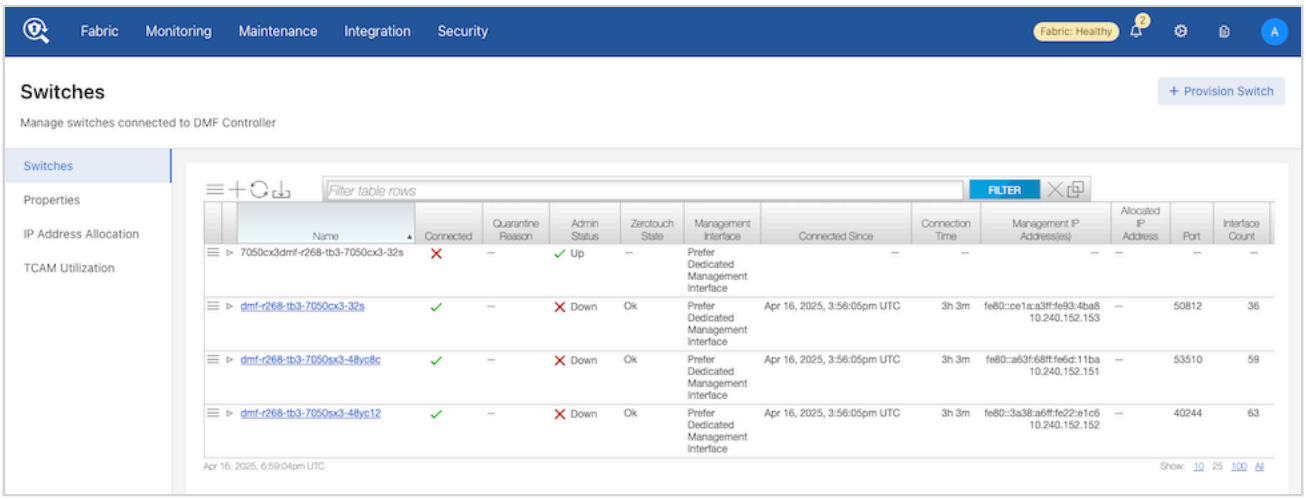
To do so, select **Admin** > **Settings** and turn off the **Switches** setting.

Figure 2-41: DMF Settings Dashboard



The Switches UI reverts to the legacy dashboard.

Figure 2-42: Legacy Switches Dashboard



2.2 Accept Additional MAC Address Formats

When configuring the MAC address of a switch, CLI commands and REST endpoints will accept a MAC address formatted as three groups of four hexadecimal digits separated by periods (e.g. `1122.3344.5566`) in addition to the already accepted form of six hexadecimal digit pairs separated by colons (e.g. `11:22:33:44:55:66`). To ensure continued compatibility for all downstream users, an address in the period-separated format is automatically translated into the colon-separated format in the configuration.

When configuring a MAC address for a switch using a method of choice (CLI, GUI, REST), enter the MAC address using the period-separated format as shown in the following example:

```
c1(config)# switch core1
c1(config-switch)# mac 1234.5678.9abc
c1(config-switch)# show this

! switch
switch core1
  mac 12:34:56:78:9a:bc
```

Considerations

The system does not maintain the format used for a given address in the configuration. Regardless of the address format entered in the configuration, it is stored and presented in the colon-separated format.

2.3 DMF Interfaces

To monitor traffic, assign a role to each of the DANZ Monitoring Fabric (DMF) interfaces, which can be of the following four types:

- **Filter Interfaces**: ports where traffic enters the DMF. Use filter interfaces to TAP or SPAN ports from production networks.
- **Delivery Interfaces**: ports where traffic leaves the DMF. Use delivery interfaces to connect to troubleshooting, monitoring, and compliance tools. These include Network Performance Monitoring (NPM), Application Performance Monitoring (APM), data recorders, security (DDoS, Advanced Threat Protection, Intrusion Detection, etc.), and SLA measurement tools.
- **Filter and Delivery interfaces**: ports with both incoming and outgoing traffic. When placing the port in loopback mode, use a filter and delivery interface to send outgoing traffic back into the switch for further processing. To reduce cost, use a filter and delivery interface when transmit and receive cables are connected to two separate devices.
- **Service interfaces**: interfaces connected to third-party services or network packet brokers, including any interface that sends or receives traffic to or from an NPB.

In addition, interfaces connected to managed service nodes and DANZ recorder nodes can be referenced in the configuration directly without assigning a role explicitly. Also, Inter-Switch Links (ISLs), which interconnect DANZ monitoring switches, are automatically detected and referred to as core interfaces.

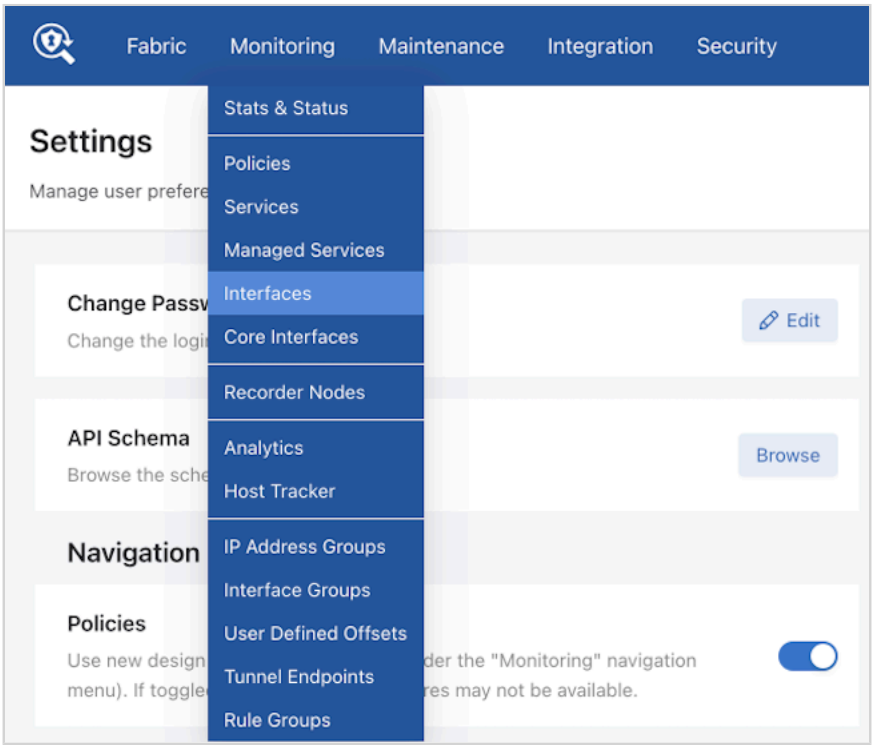
This chapter describes configuring DMF interfaces.

2.3.1 Overview

While preserving the information from the previous DMF version, the updated **DMF Interfaces** user interface introduces a new layout, design, and enhanced functionalities for improved interface viewing and monitoring for easy troubleshooting.

Select **Monitoring** > **Interfaces** to access **DMF Interfaces**.

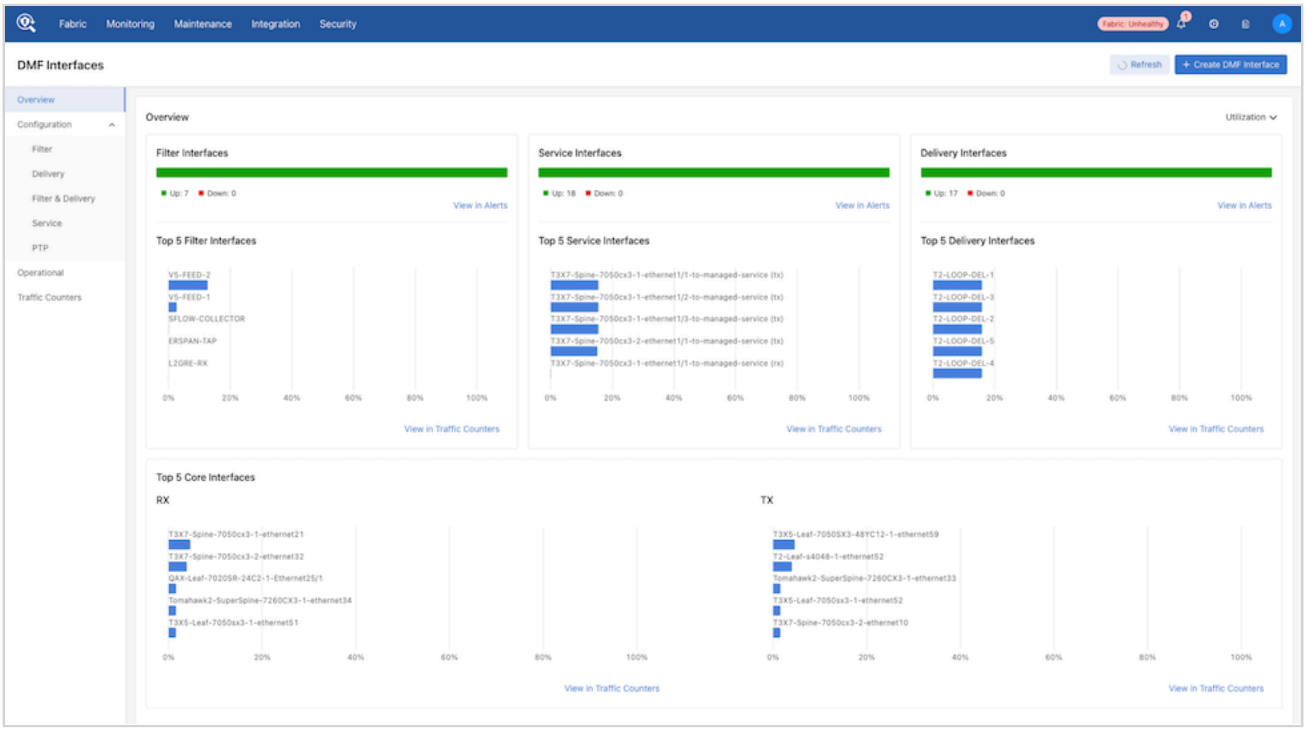
Figure 2-43: Monitoring Interfaces



Overview Dashboard

Overview is the default landing page for **DMF Interfaces**.

Figure 2-44: Overview Dashboard



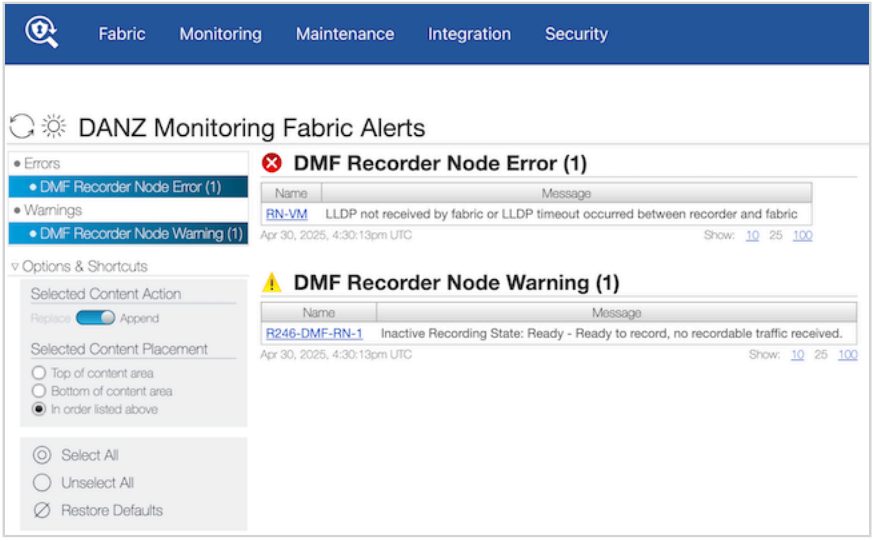
DMF displays interface status in:

- **Filter Interfaces**
- **Service Interfaces**
- **Delivery Interfaces**

Each interface displays an **Up** and **Down** count.

Select **View Alerts** to navigate to the **Alerts** dashboard for more details on the down interfaces.

Figure 2-45: Interface Alerts



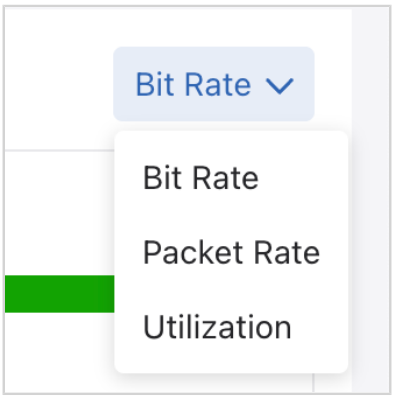
DMF displays interface metrics in:

- **Top 5 Filter Interfaces**
- **Top 5 Service Interfaces**
- **Top 5 Delivery Interfaces**
- **Top 5 Core Interfaces**
  - **RX** - Receive
  - **TX** - Transmit

Select **View in Traffic Counters** to navigate to **Traffic Counters** (described later in this section).

Select the **Utilization**, **Bit Rate**, or **Packet Rate** (Unit) drop-down to switch between utilization, bit rate, and packet rate data. The **Overview** widgets update with the selected unit type. The selection persists until changed.

Figure 2-46: Units

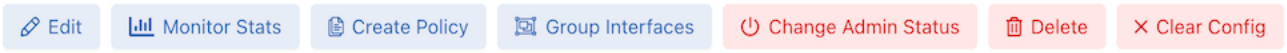


2.3.2 Configuration

The **Configuration** dashboard displays tabular configuration information for the following:

- **Filter Interfaces**
- **Delivery Interfaces**
- **Filter & Delivery Interfaces**
- **Service Interfaces**
- **PTP Interfaces**

Action Buttons (Example - Filter Interfaces)



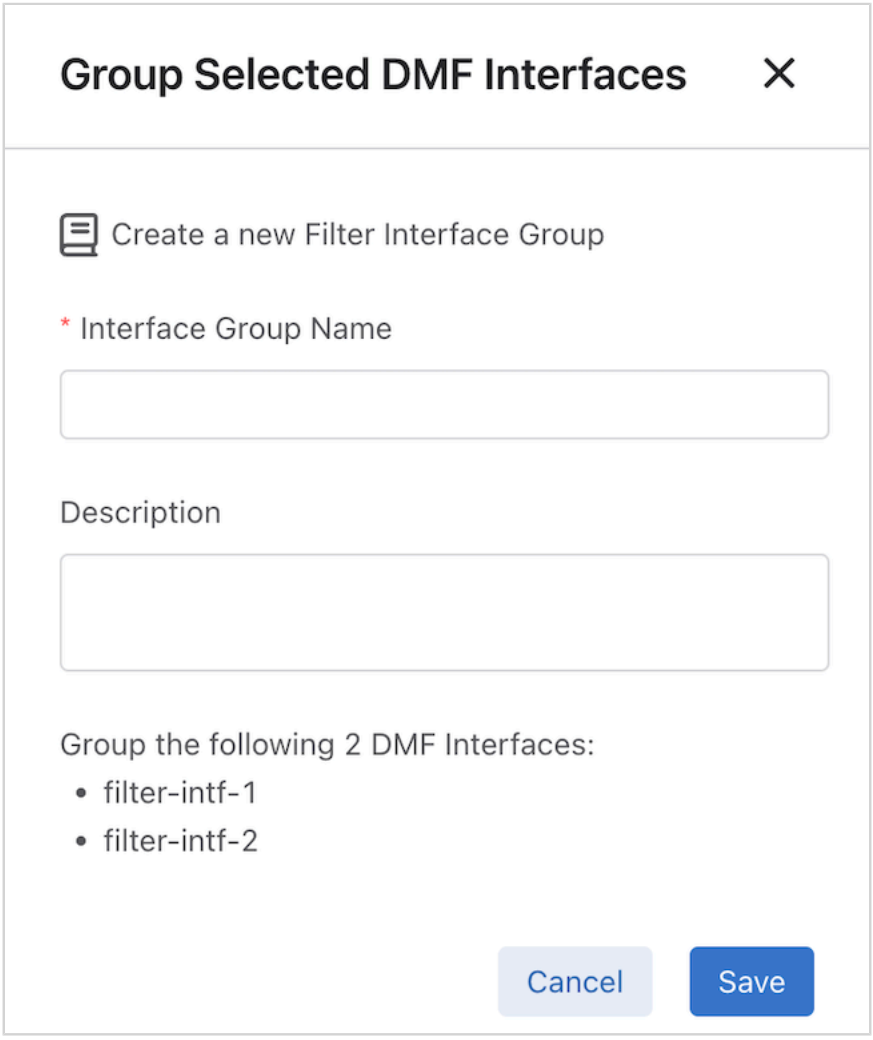
Edit DMF Interface: **Edit** opens the **Edit DMF Interface** window, which is pre-populated with the interface information.

- When selecting multiple rows **Edit** is disabled.

Monitor Stats opens the **Traffic Counters** (described later in this section) window.

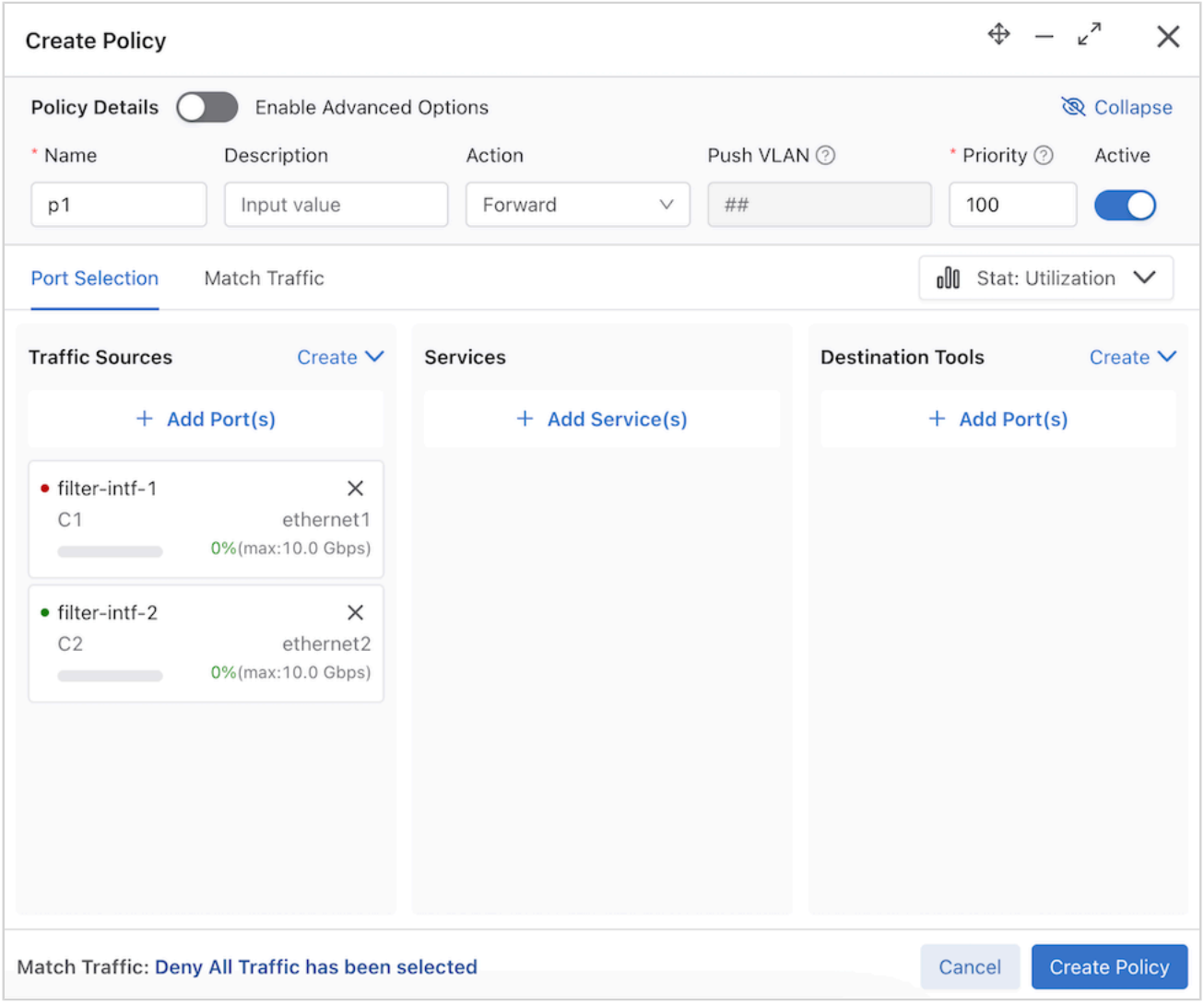
Group Interfaces opens the **Group Selected DMF Interfaces** configuration window, which is pre-populated with the selected interfaces.

Figure 2-47: Group Selected DMF Interfaces



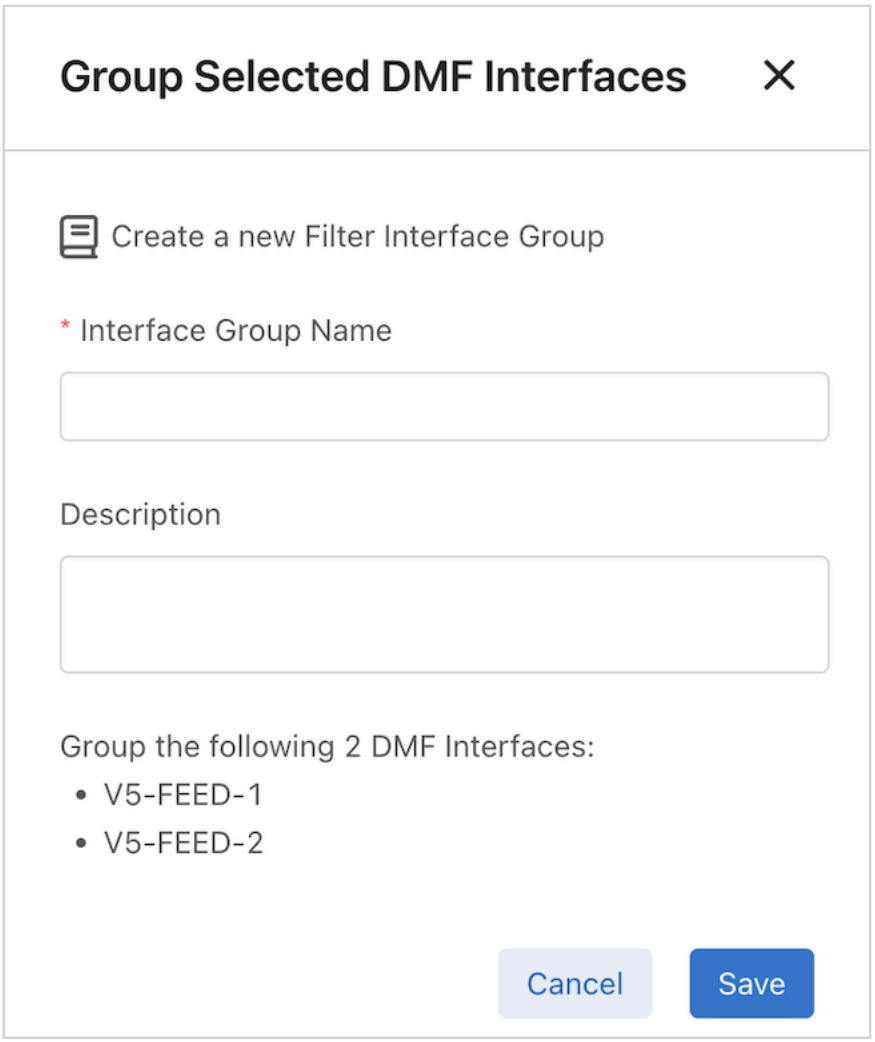
Create Policy opens the Create Policy configuration window for the selected interfaces.

Figure 2-48: Create Policy

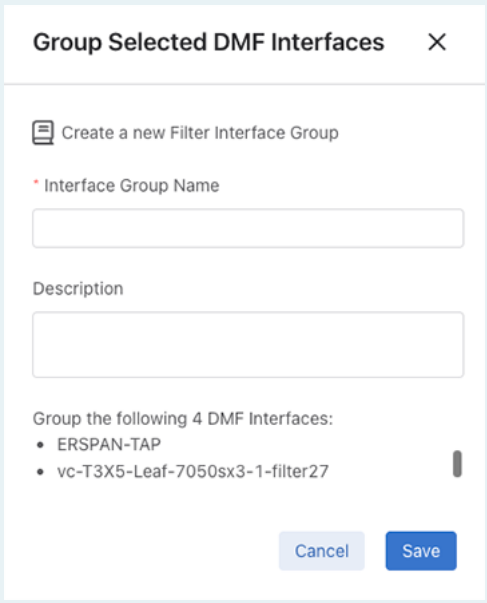


Group Interfaces opens the **Group Selected DMF Interfaces** window for the selected interfaces.

Figure 2-49: Group Selected DMF Interfaces

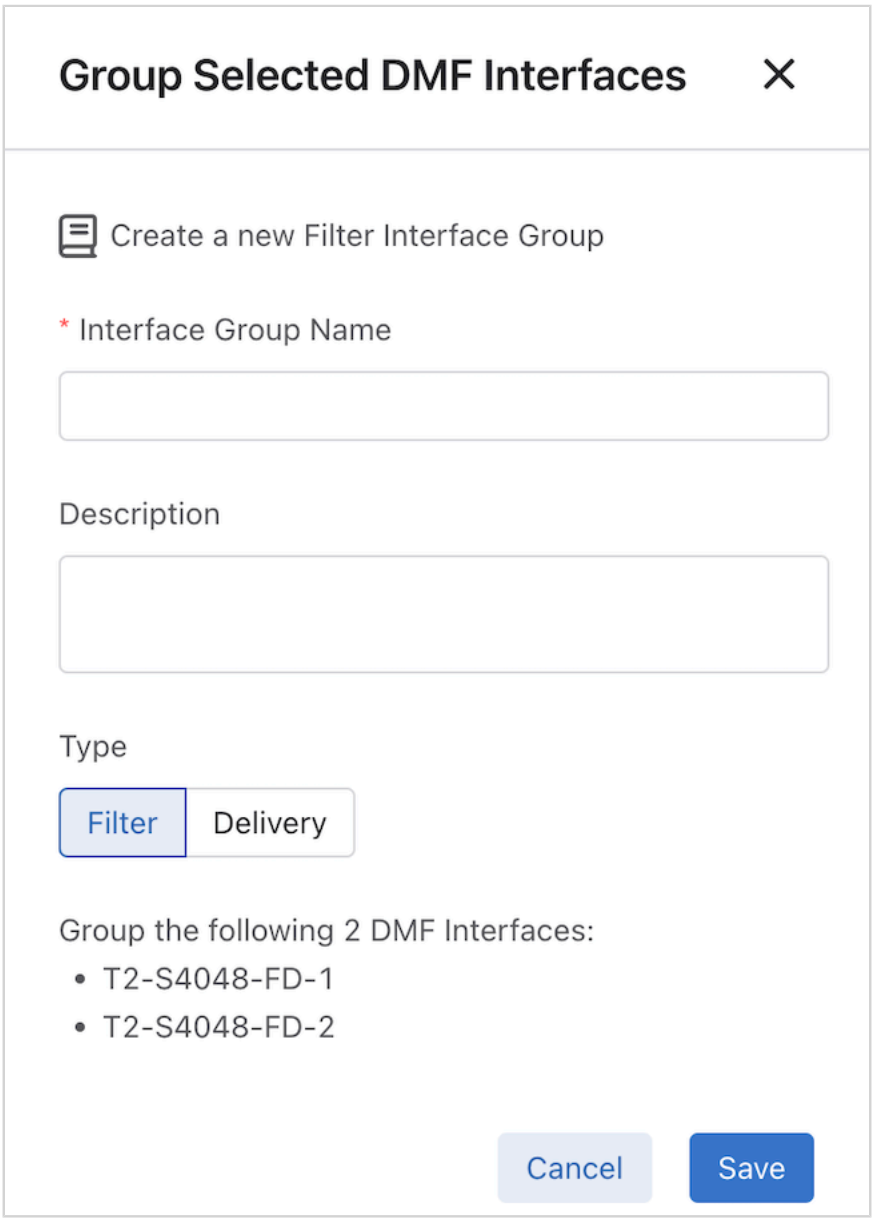


Tip: When grouping more than two interfaces, hover over a name and use the scroll feature to view the interface names.



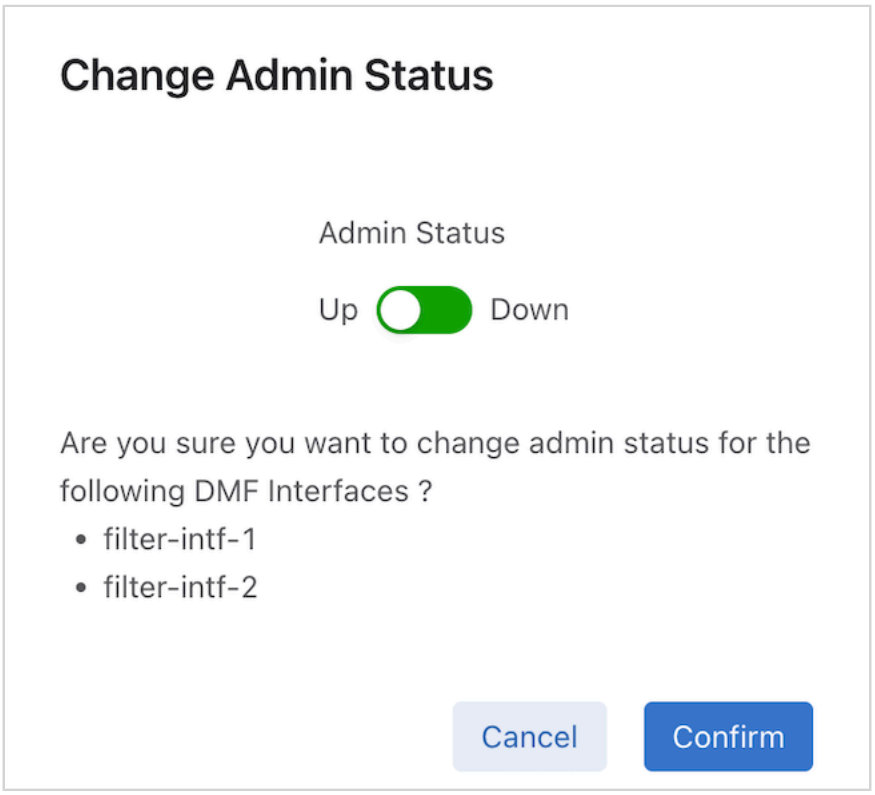
There is an option to select the type of role to assign to the **Filter** and **Delivery** interfaces.

Figure 2-50: Filter and Delivery Type



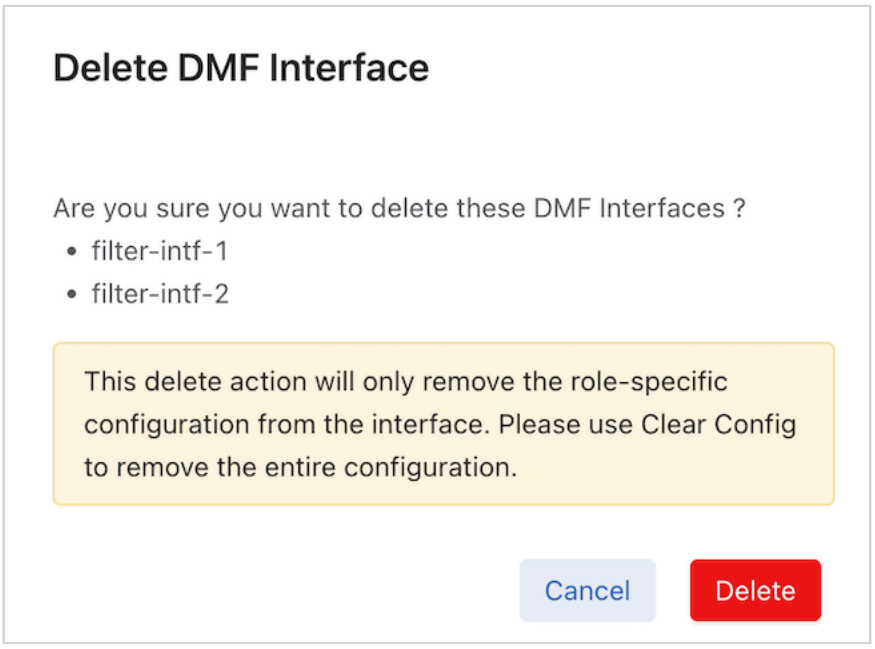
Change Admin Status changes the Admin Status (**Up** or **Down**).

Figure 2-51: Change Admin Status



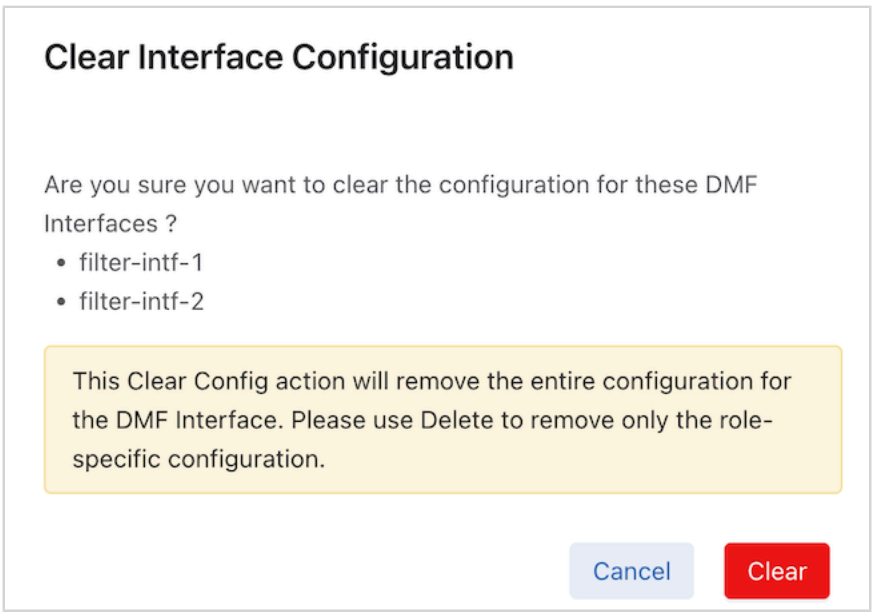
Delete (Delete DMF Interface) removes a DMF Interface role-specific configuration from an interface. It does not remove the entire configuration.

Figure 2-52: Delete DMF Interface

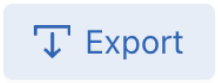


Clear Config (Clear Interface Configuration) removes the entire configuration for the DMF Interface.

Figure 2-53: Clear Interface Configuration

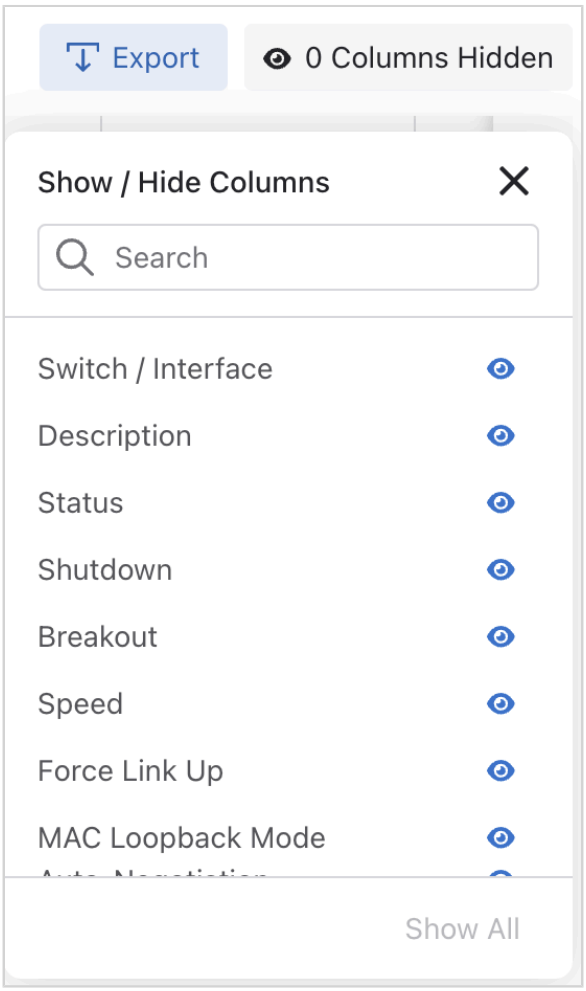


Export saves interface data content in a CSV or JSON format in the browser's download folder.



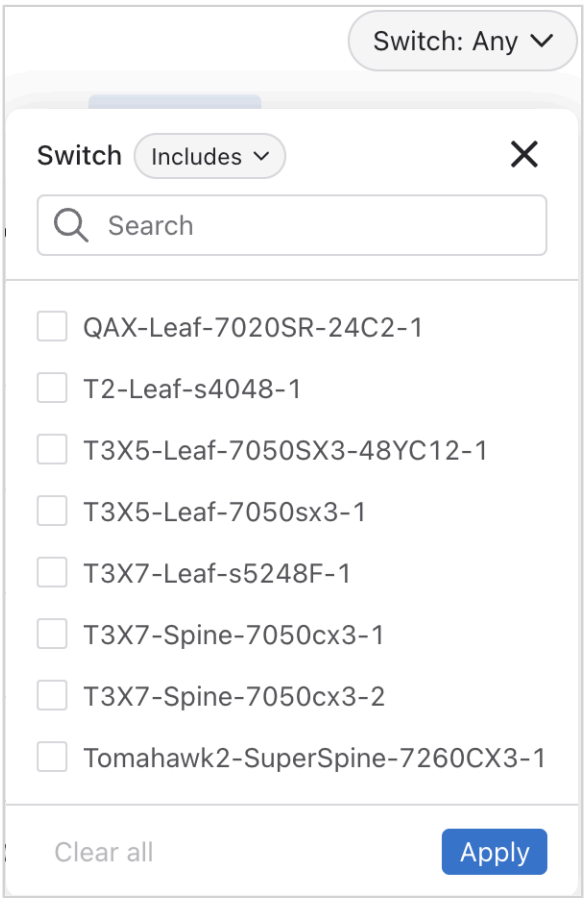
Show / Hide Columns determine the information displayed in the dashboard and are user-selectable. Selections persist until changed.

Figure 2-54: Show / Hide Columns



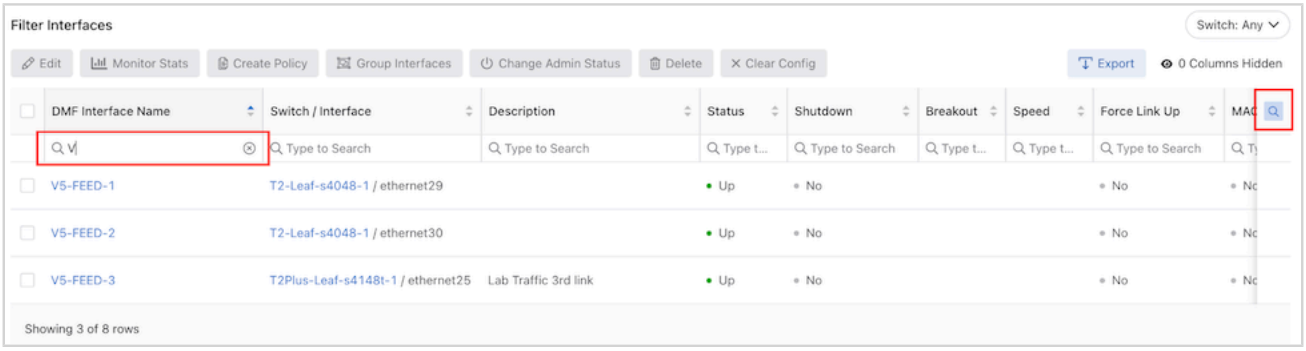
Switch filters by switch name using **Includes** and **Excludes**.

Figure 2-55: Switch Selection



Enable **Search** functionality by column using the search icon. **Type to Search** appears in each column. Enter the required search criteria.

Figure 2-56: Search

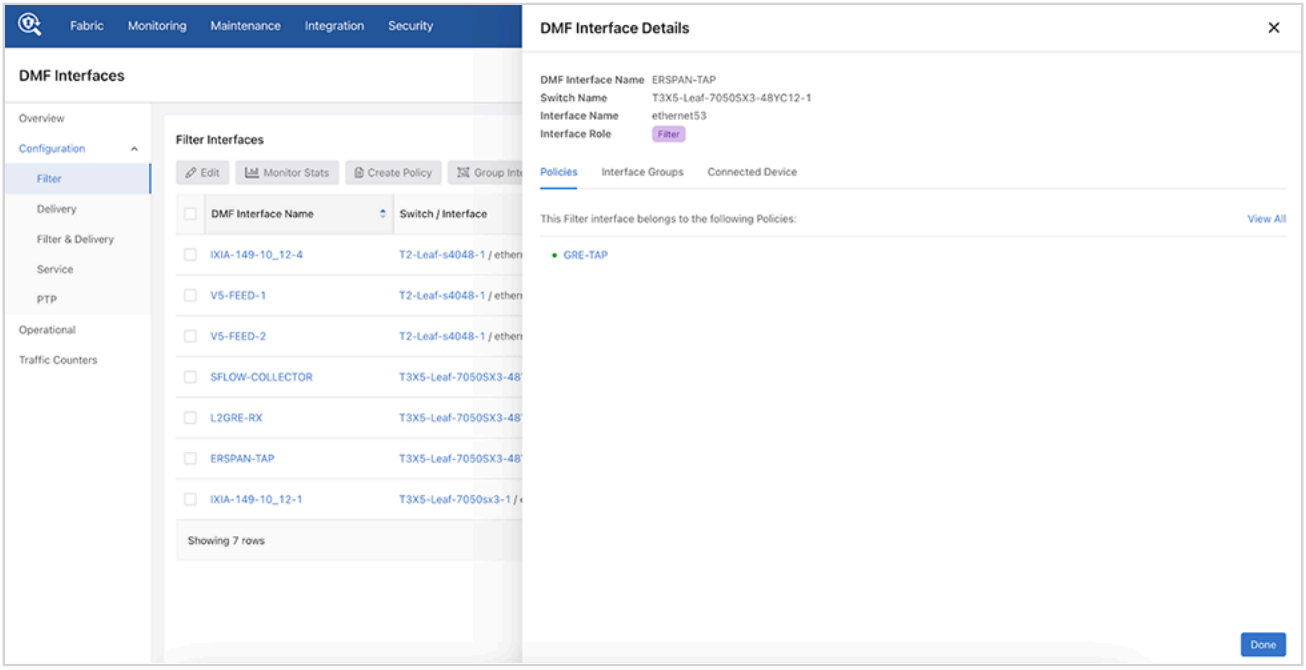


Interface Details View

Selecting a **DMF Interface Name** entry opens the **DMF Interface Details** window for the **Filter**, **Delivery**, and **Filter & Delivery** interfaces.

The **DMF Interface Details** view displays interface information associated with **Policies**, **Interface Groups**, and **Connected Devices**. It supports configuring VLAN Preservation and Egress Filtering Rules, as shown in the following examples:

Figure 2-57: DMF Interface Details - Policies



Hovering over a filter interface configured in a policy displays further status and configuration information:

Figure 2-58: Filter Interface - Policy

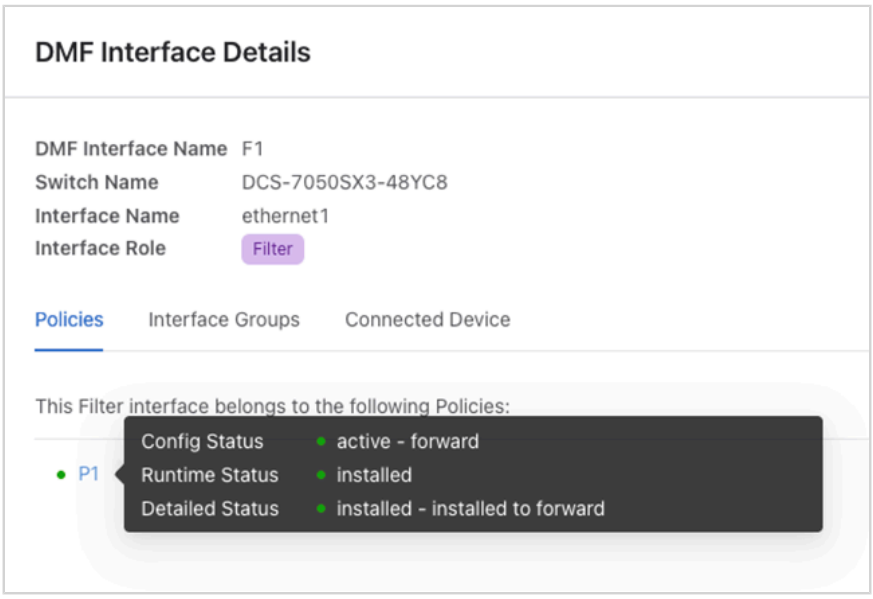


Figure 2-59: DMF Interface Details - Interface Groups



Figure 2-60: DMF Interface Details - Connected Devices

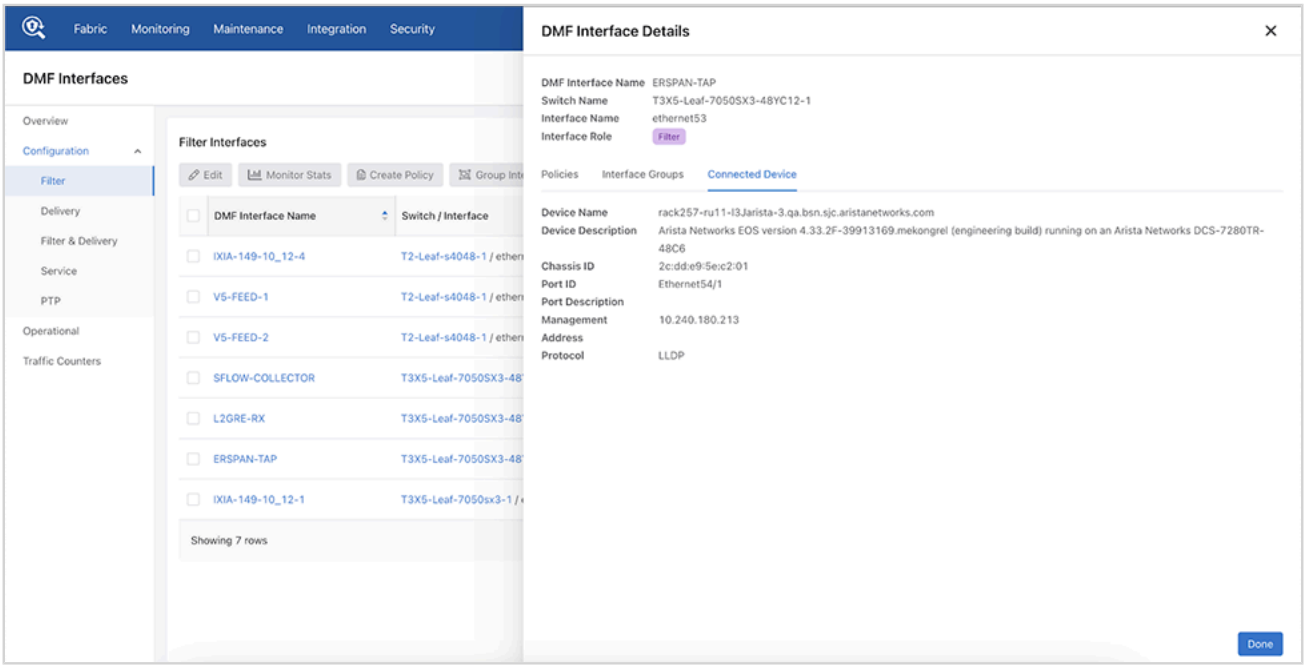
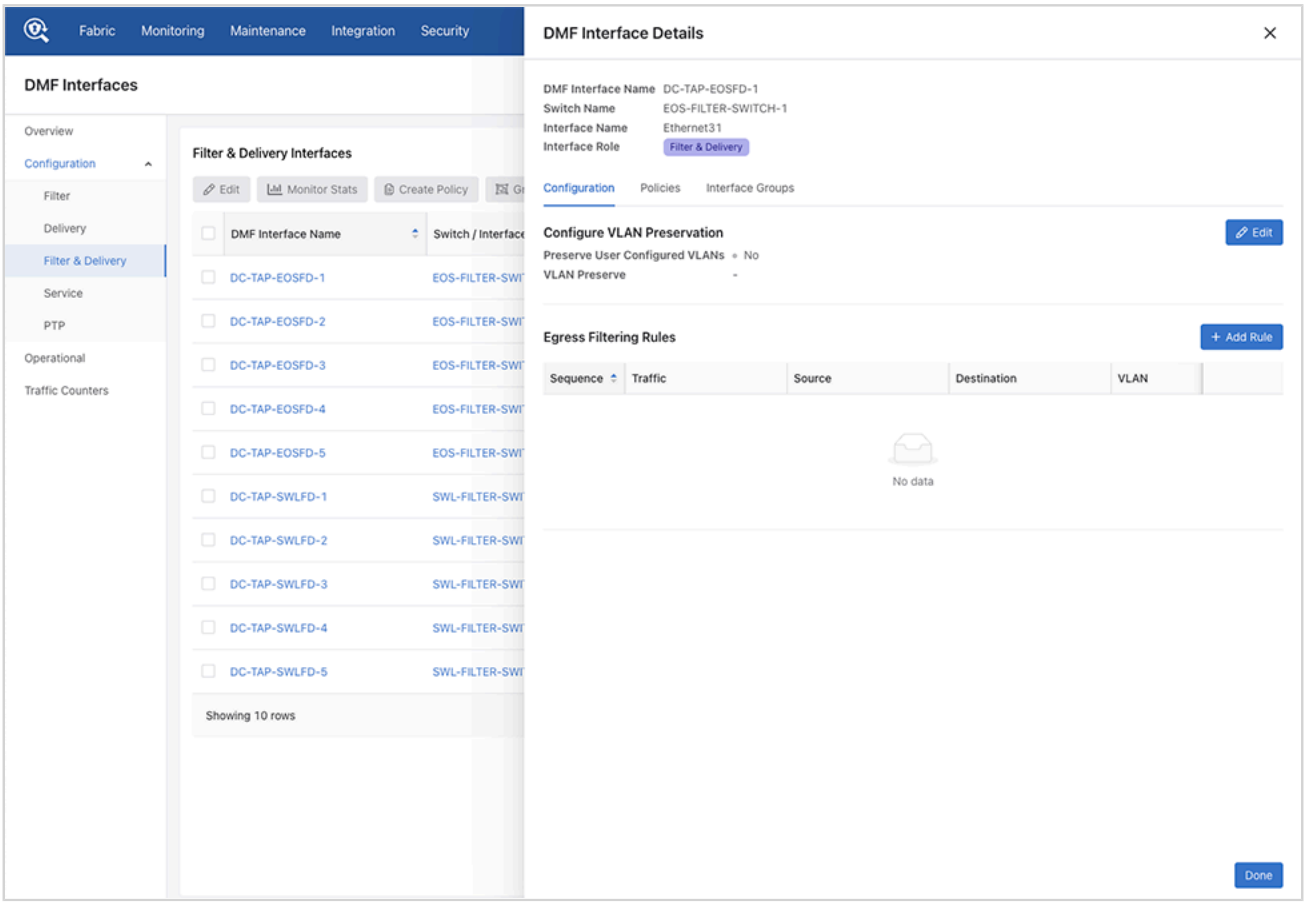


Figure 2-61: DMF Interface Details - Filter & Delivery



Select **Edit** to configure **VLAN Preservation**. Enable the **Preserve User Configured VLANs** checkbox, use the drop-down menu, and select the required VLAN numbers to preserve.

Figure 2-62: VLAN Preservation

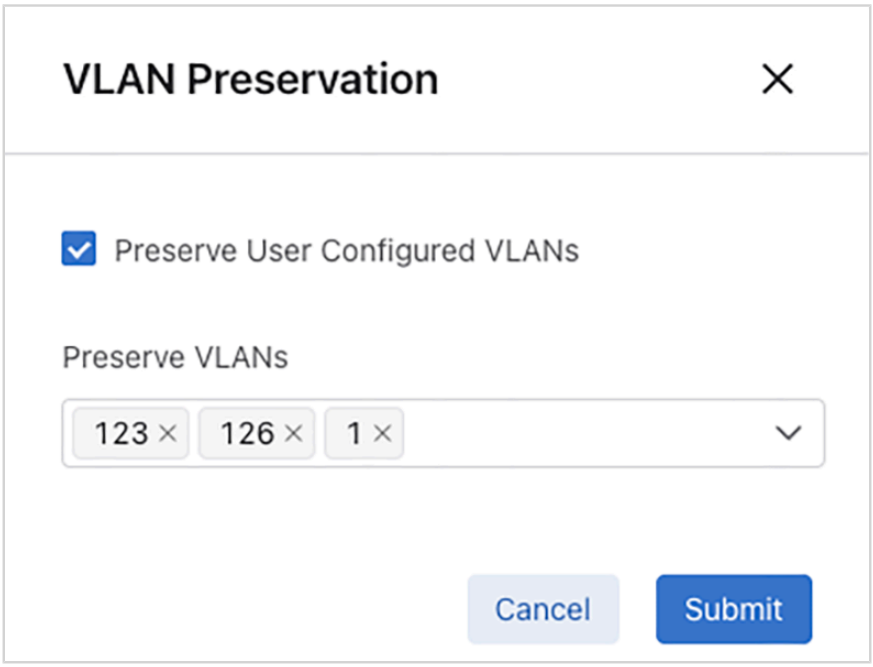
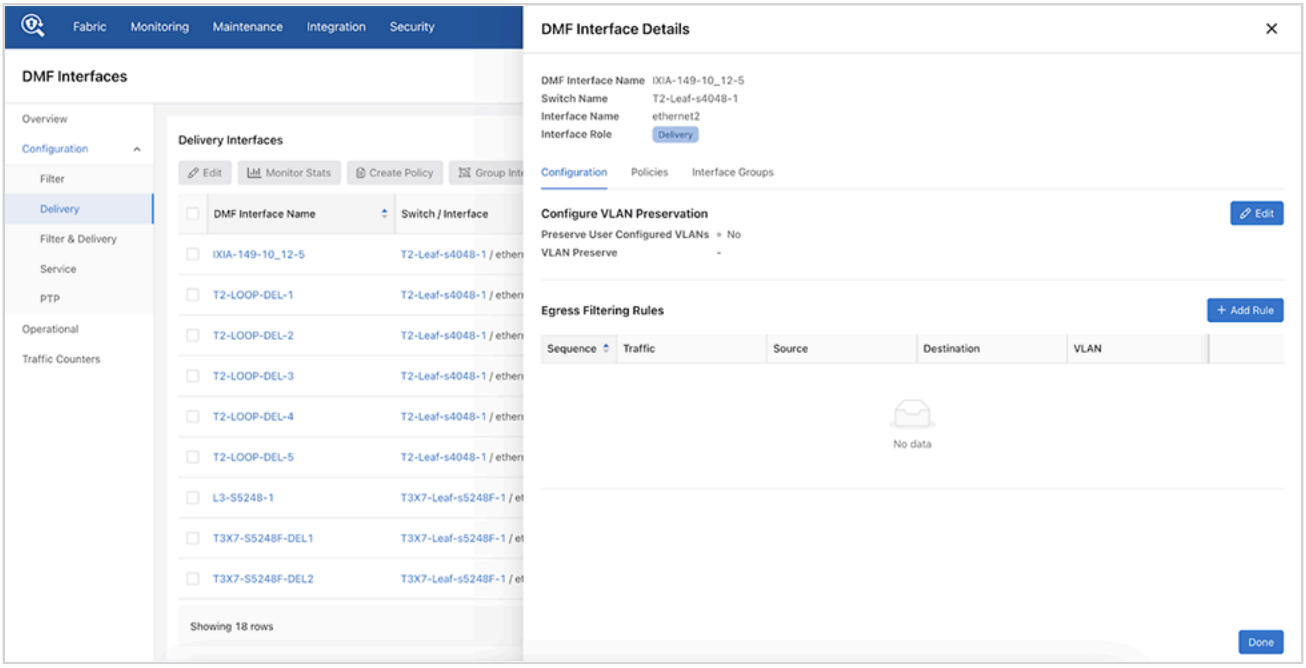
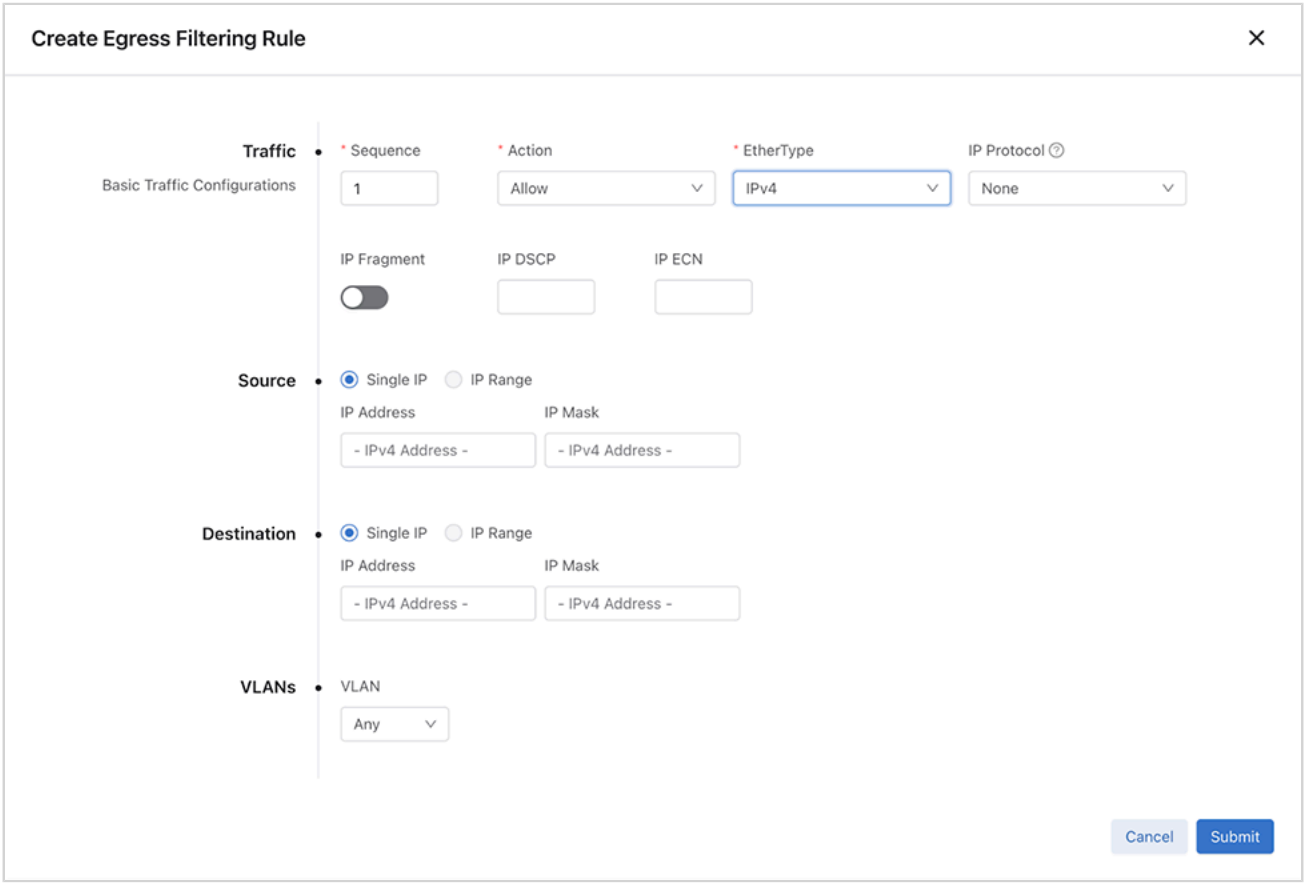


Figure 2-63: DMF Interface Details - Egress Filtering Rules



Select **+ Add Rule Edit** to create an egress filtering rule configure egress filtering. Enter the required information in the **Create Egress Filtering Rule** configuration window.

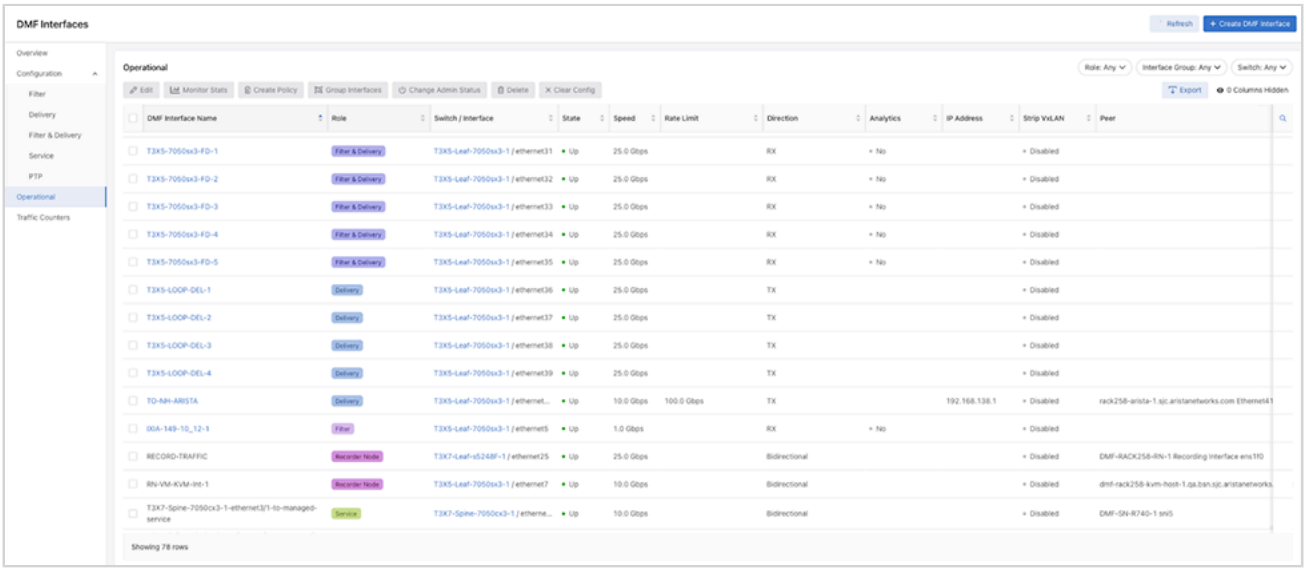
Figure 2-64: Create Egress Filtering Rule



2.3.3 Operational

The **Operational** dashboards displays functional information for each interface in tabular format.

Figure 2-65: Operational Dashboard



Action Buttons

These work in the same manner as the action buttons in **Configuration**.

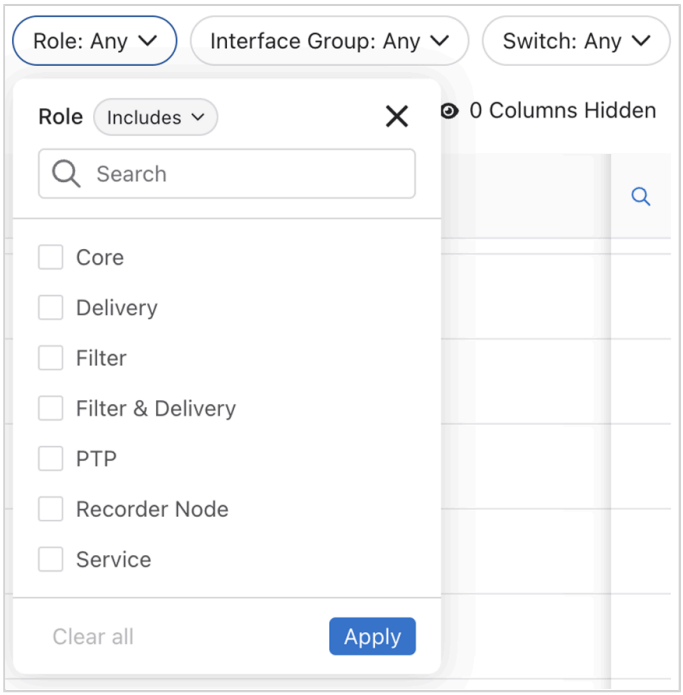
- **Edit**
- **Monitor Stats**
- **Create Policy**
- **Group Interfaces**
- **Change Admin Status**
- **Delete**
- **Clear Config**
- **Export**
- **Show / Hide Columns**

Search

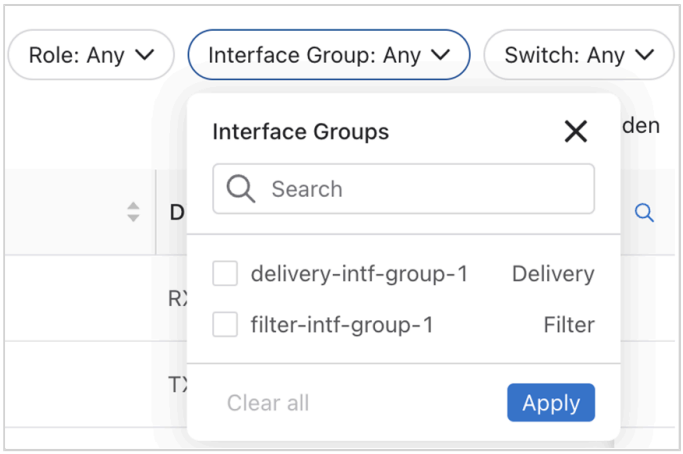
The **Search** function works in the same manner as the search function in **Configuration**.

Filters

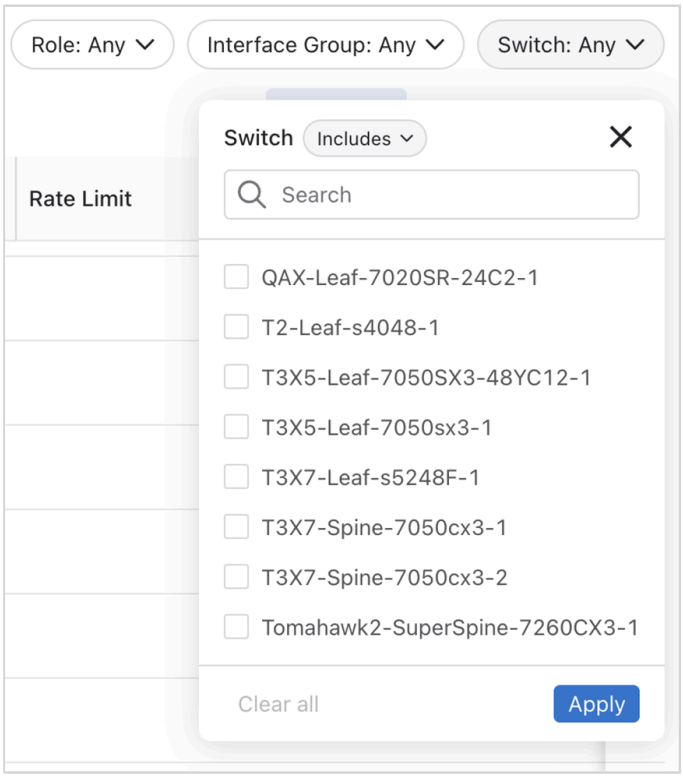
- **Filter by Role** - Search or select from the list and filter by **Includes** or **Excludes**.



- **Filter by Interface Groups** - Search or select from the list.



- **Filter by Switch Name** - Search or select from the list and filter by **Includes** or **Excludes**.



Select **Apply** to begin filtering.

2.3.4 Traffic Counters

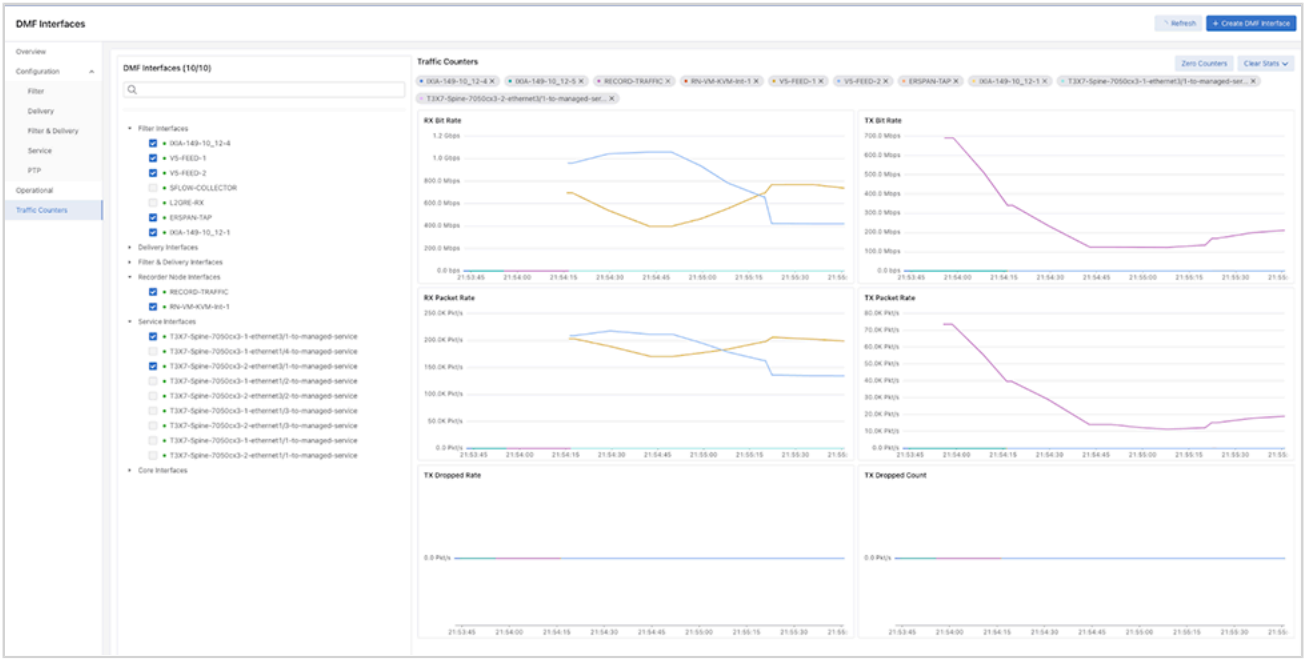
The **Traffic Counters** dashboard display the utilization statistics for the interfaces. Six time-series charts display the following stats:

- **RX Bit Rate**
- **TX Bit Rate**
- **RX Packet Rate**
- **TX Packet Rate**
- **TX Dropped Rate**
- **TX Dropped Count**

The charts are updated every 10 seconds and display each interface's utilization.

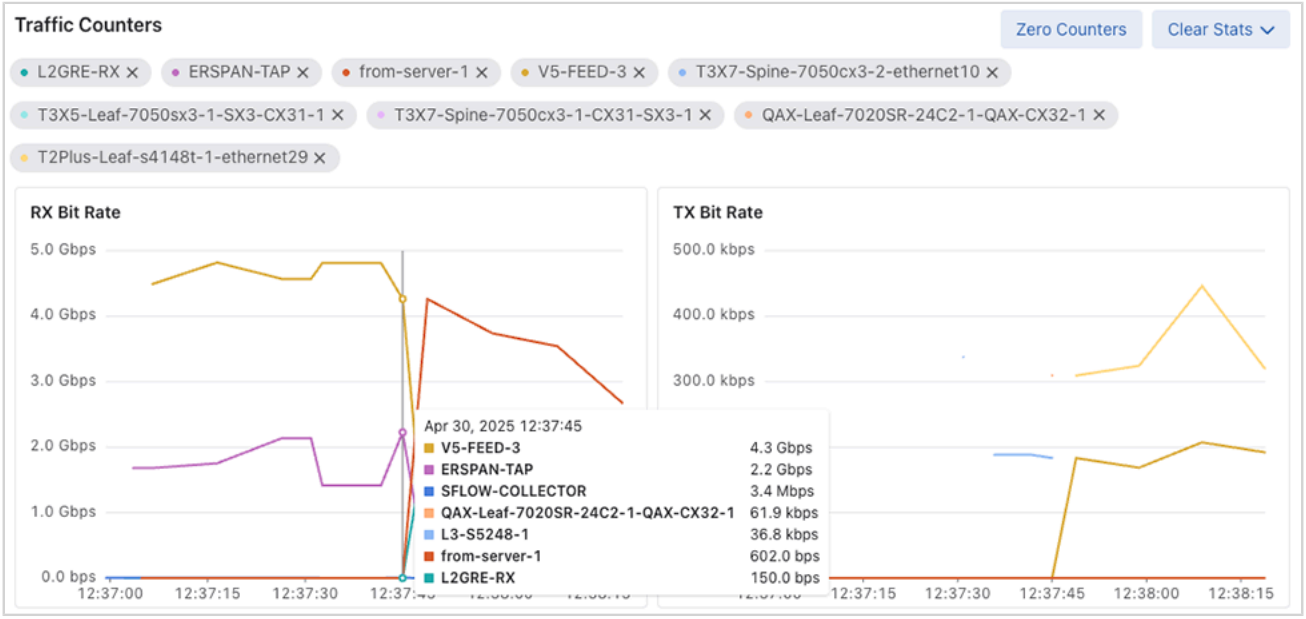
You can select up to 10 interfaces at a time, and selecting a new interface re-initializes the charts, and polling begins for new interface utilization data.

Figure 2-66: DMF Interfaces - Traffic Counters

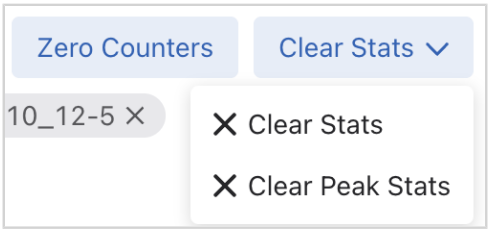


Hovering over and selecting graph plot displays bit rate information.

Figure 2-67: Traffic Counter Rollover



Action Controls



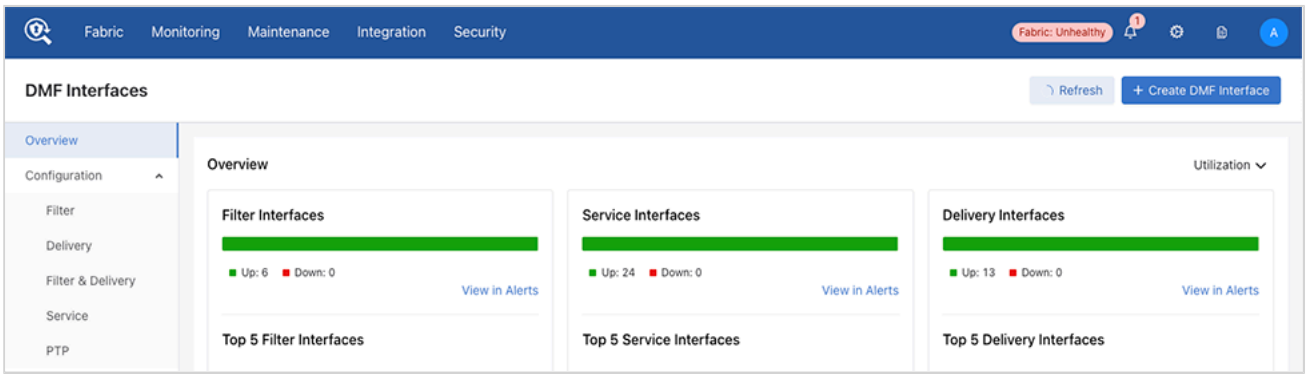
- **Zero Counters**: Resets counters for the selected Interfaces in this UI session.
- **Clear Stats**: Clear stats for the selected Interfaces.
- **Clear Peak Stats**: Clear peak stats for the selected Interfaces.

2.3.5 Configure a DMF Interface

To use the DANZ Monitoring Fabric (DMF) GUI to configure a fabric interface as a **Filter**, **Service**, **Delivery**, **Filter & Delivery**, or **PTP** interface, perform the following steps:

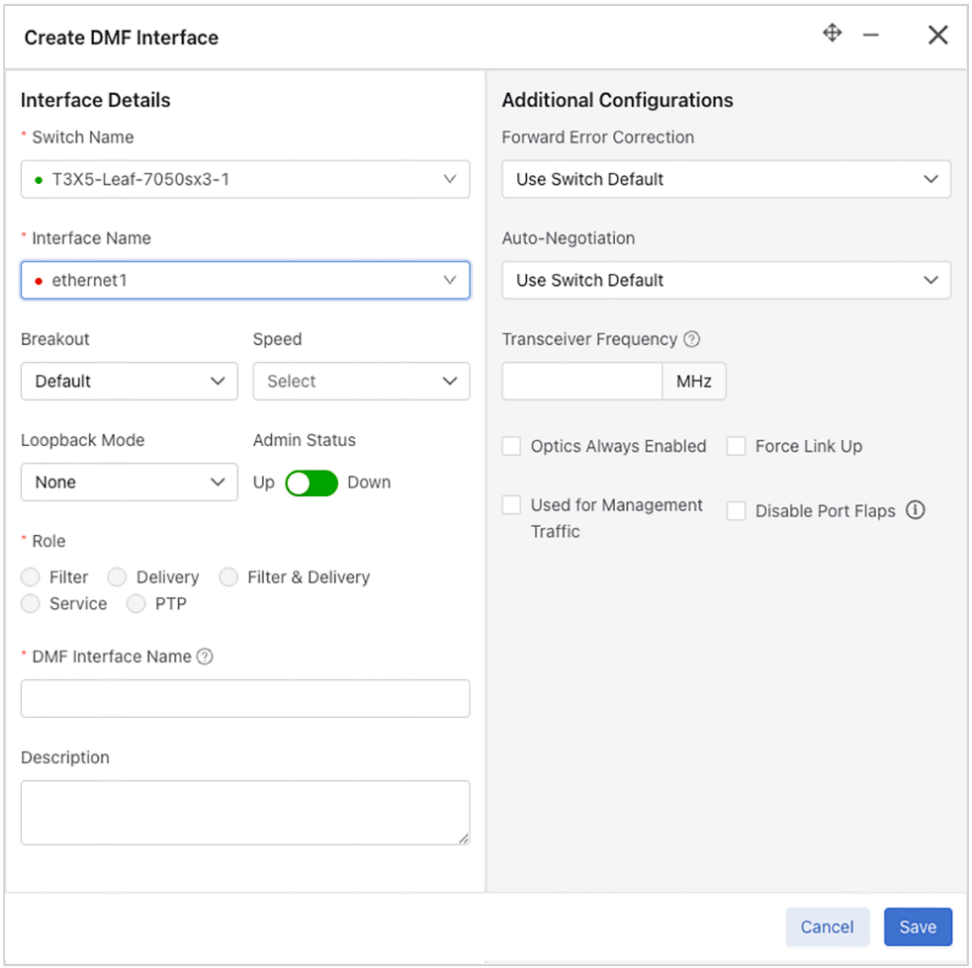
1. Select **Monitoring > Interfaces** from the main menu to display the DMF interfaces.

   Figure 2-68: DMF Interfaces



2. Select the **+ Create DMF Interface** to configure a new interface.

   Figure 2-69: Create Interface



3. Select the **Switch Name** and **Interface Name**.
4. Define the **Role** for the interface.
   - **Filter**
   - **Delivery**
   - **Filter & Delivery**
   - **Service**
   - **PTP**
5. Enter a **DMF Interface Name** and a **Description** (optional).
6. If required, enter an **IPv4 Address** for receiving IP datagram traffic in a **Filter** or **Delivery** role.
7. Complete the settings in **Additional Configurations**, as required, for the selected role.
   - **Forward Error Correction** - Select from a drop-down list of FEC codes.
   - **Auto-Negotiation** - **Enable** or **Disable**
   - **Transceiver Frequency**
   - **Optics Always Enabled** (checkbox)
   - **Force Link Up** (checkbox)
   - **Used for Management Traffic** (checkbox)
   - **Disable Port Flaps** (checkbox) - If the **Error Disable** feature is enabled globally, checking this input will disable the **Error Disable** feature on this interface temporarily.
8. Select **Save** to save the configuration.

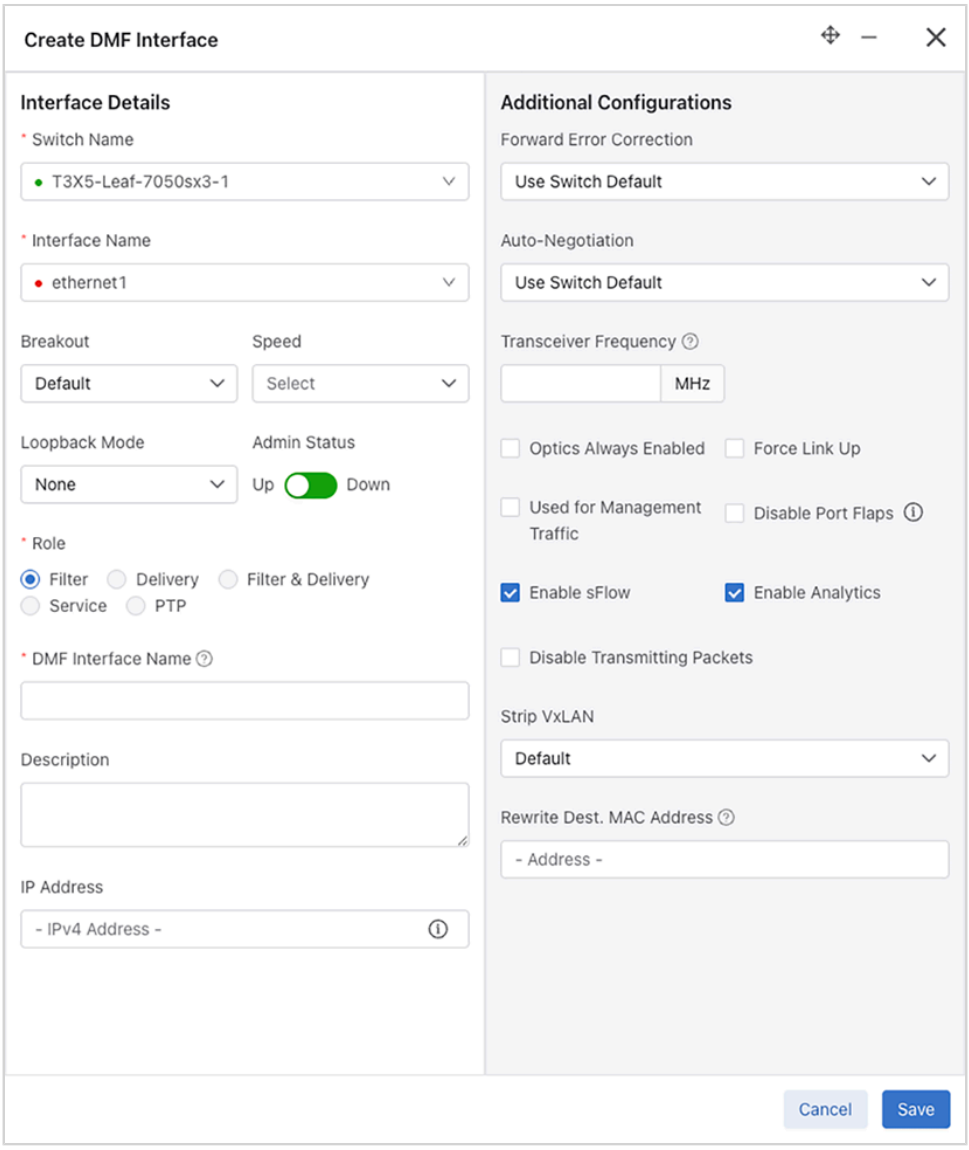
The following images depict the specific **Additional Configurations** settings for each role.

Filter Interface Role

Additional configuration settings for the **Filter** role include:

- **Enable sFlow** (checkbox) - Enabled by default.
- **Enable Analytics** (checkbox) - Enabled by default. For information about Analytics, refer to the *Analytics Node User Guide*.
- **Disable Transmitting Packets** (checkbox)
- **Strip VXLAN** - **Enable** or **Disable**
- **Rewrite Dest. MAC Address** - The provided MAC address will overwrite the destination MAC address of packets received on this interface.

Figure 2-70: Role - Filter Interface



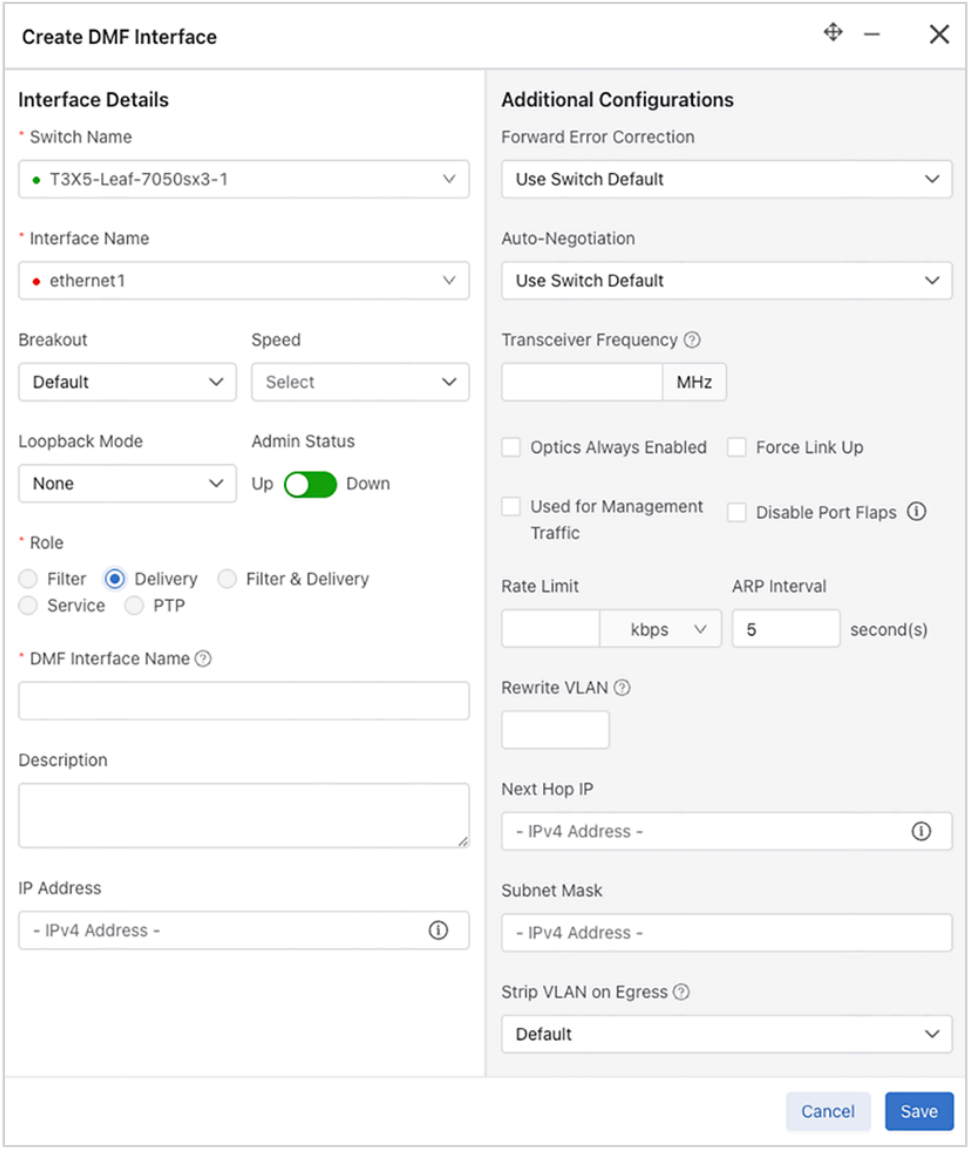
Delivery Interface Role

Additional configuration settings for the **Delivery** role include:

- **Rate Limit** - Enter a value and select bit rate kbps, Mbps, or Gbps.
- **ARP Interval** - Default is 5 seconds.
- **Rewrite VLAN** - The vlan tag to push/rewrite for this interface.
- **Next Hop ID** - **IPv4 Address**. Next hop IP that is reachable via delivery interface with local mask.
- **Subnet Mask** - **IPv4 Address**
- **Strip VLAN on Egress** - **Default**, **None**, **One**, **Two**, or **Second**.

- **VLAN Preservation** (checkbox) - Enabled by default.

Figure 2-71: Role - Delivery Interface

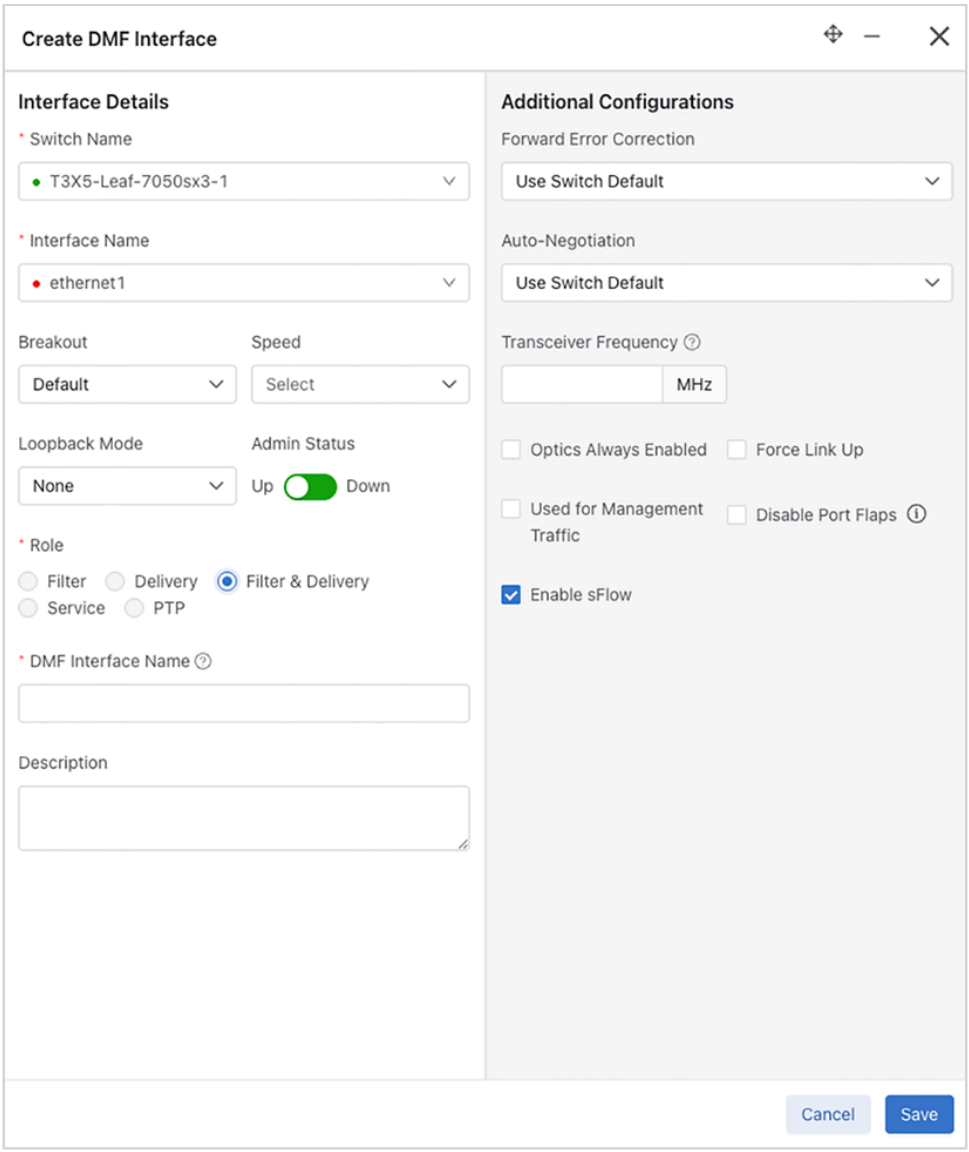


Filter & Delivery Interface Role

Additional configuration settings for the **Filter & Delivery** role include:

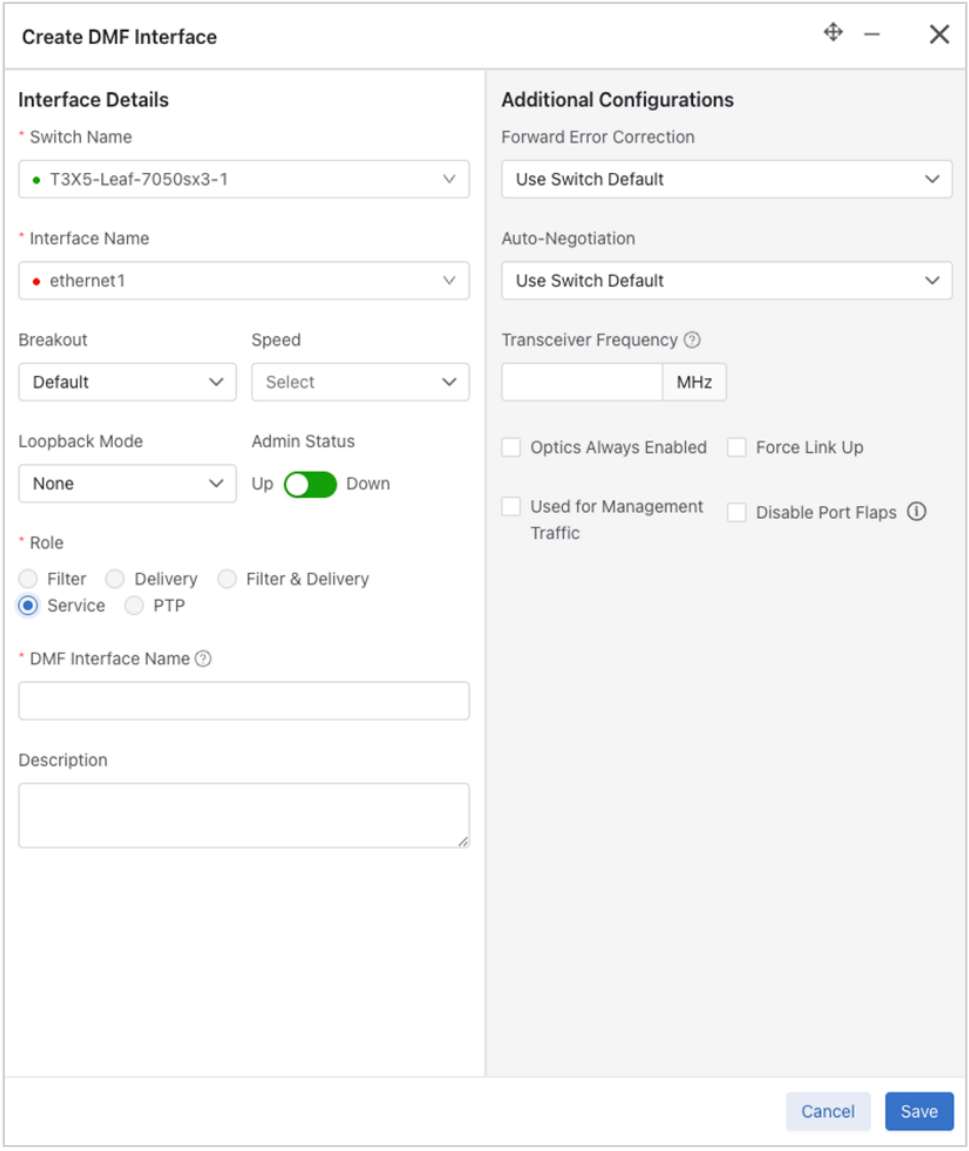
- **Enable sFlow** (checkbox) - Enabled by default.

Figure 2-72: Role - Filter & Delivery Interface



Service Interface Role

Figure 2-73: Role - Service Interface

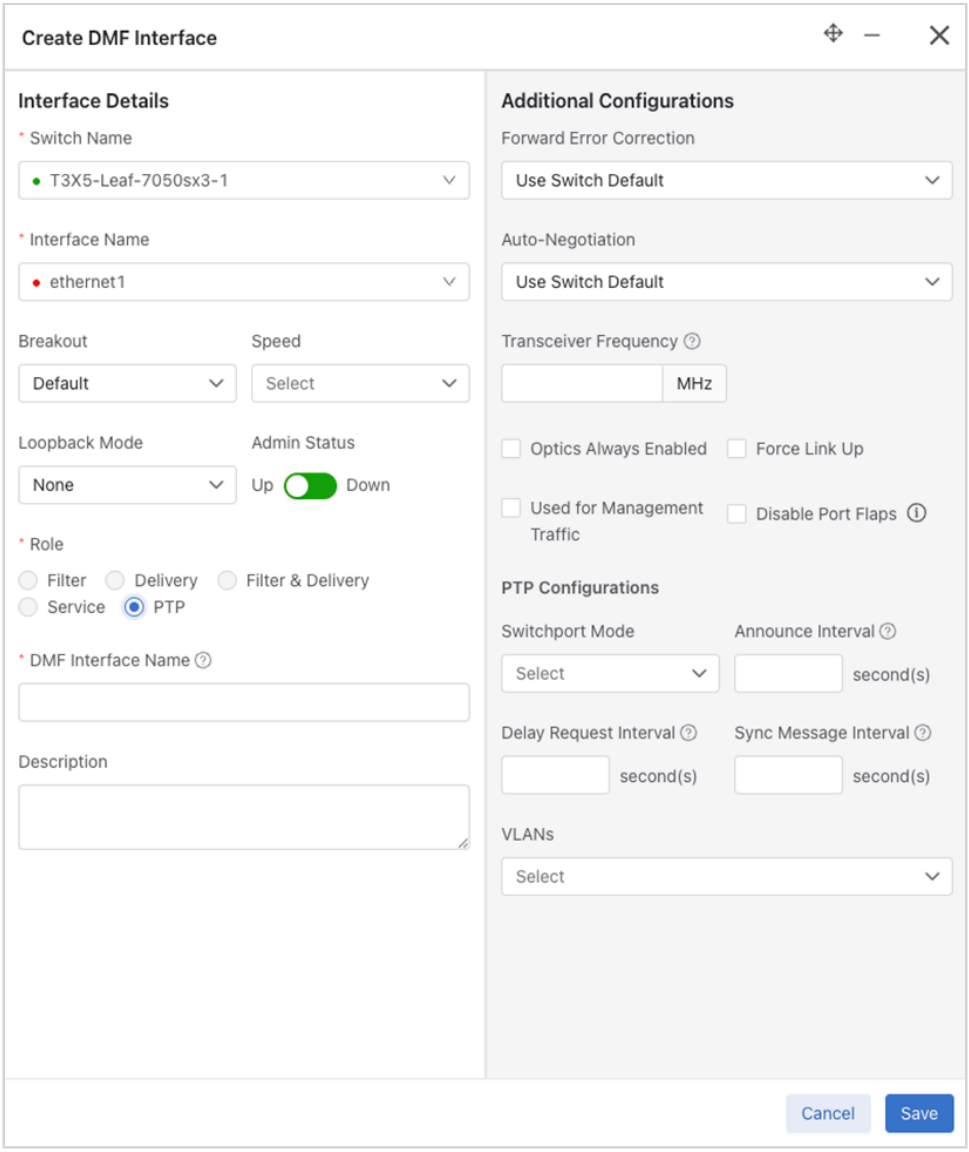


PTP Interface Role

Settings for PTP configurations:

- **Switchport Mode** - **Access**, **Routed**, or **Trunk**.
- **Announce Interval** - Set PTP announce interval between messages (-3 to 4).
- **Delay Request Interval** - Set PTP delay request interval between messages (-7 to 8).
- **Sync Message Interval** - Set PTP sync message interval between messages (-7 to 3).
- **VLANS** - Select from a drop-down list of VLANs.

Figure 2-74: Role - PTP Interface



2.3.6 Identify a Filter Interface using Destination MAC Rewrite

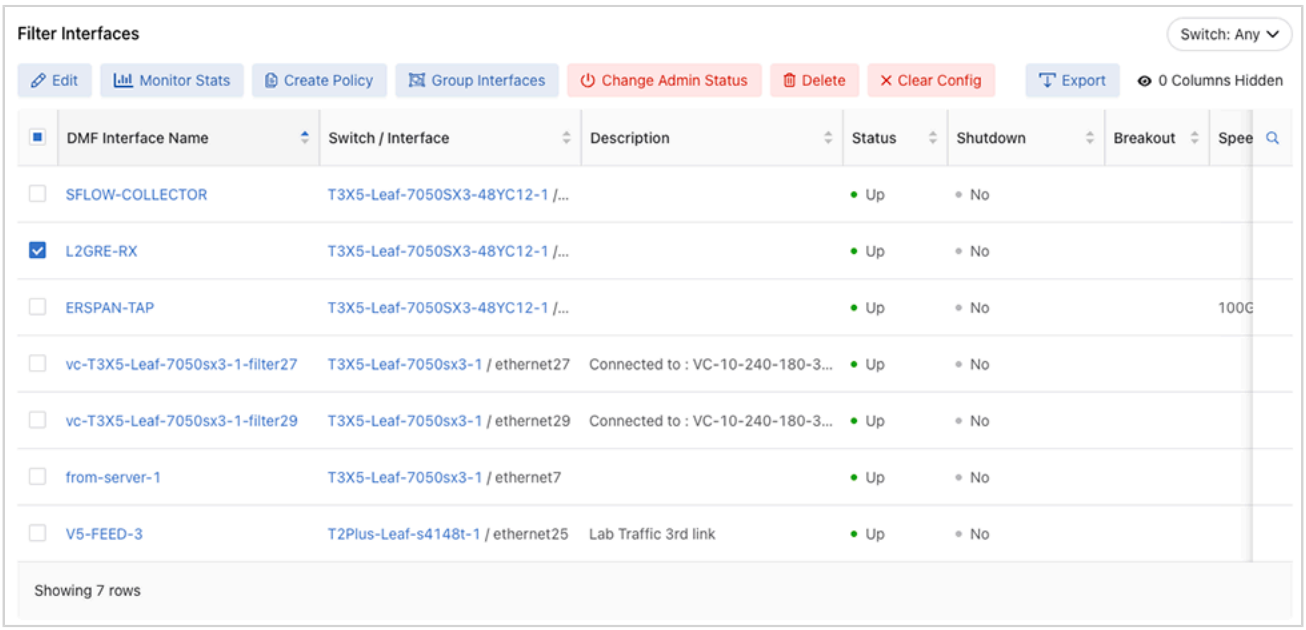
In the UI, configure the **Rewrite Dest. MAC Address** for a **Filter Interface** using one of the two workflows detailed below.

Workflow one uses the **Monitoring** > **Interfaces**, while the second uses the **Fabric** > **Interfaces**. To use the second workflow, proceed to step 6, detailed below.

Workflow One

Navigate to **Monitoring** > **Interfaces** > **Configuration** > **Filter**.

Figure 2-75: Filter Interfaces

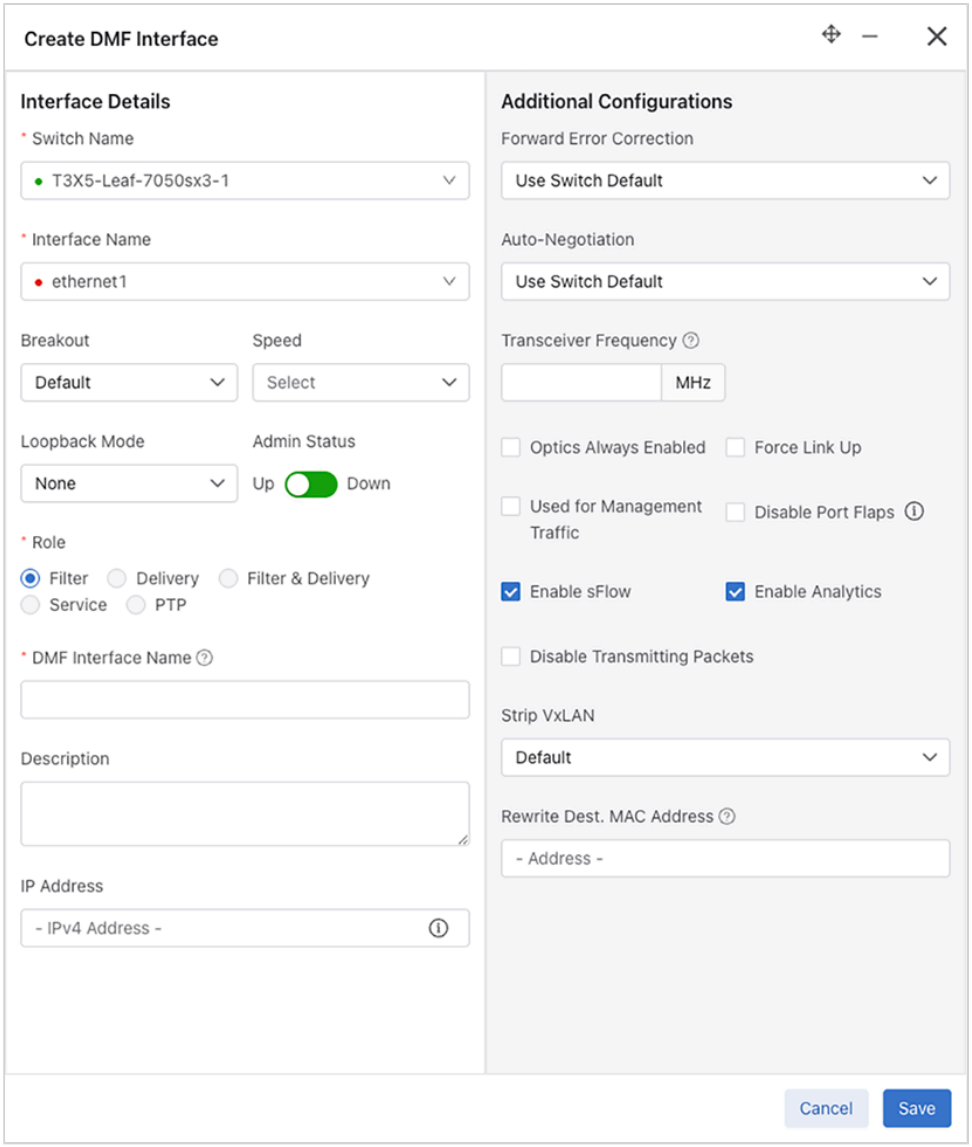


Filter Interfaces

1. To create a Filter interface refer to Configure a DMF Interface Filter role.
2. Enter the required **Rewrite Dest. MAC Address**. The provided MAC address will overwrite the destination MAC address of packets received on this interface.

3. Select the **Role** > **Filter** for the desired interface and use the **Rewrite Dest.MAC Address** input to configure the MAC address to override.

Figure 2-76: Role - Filter Interface

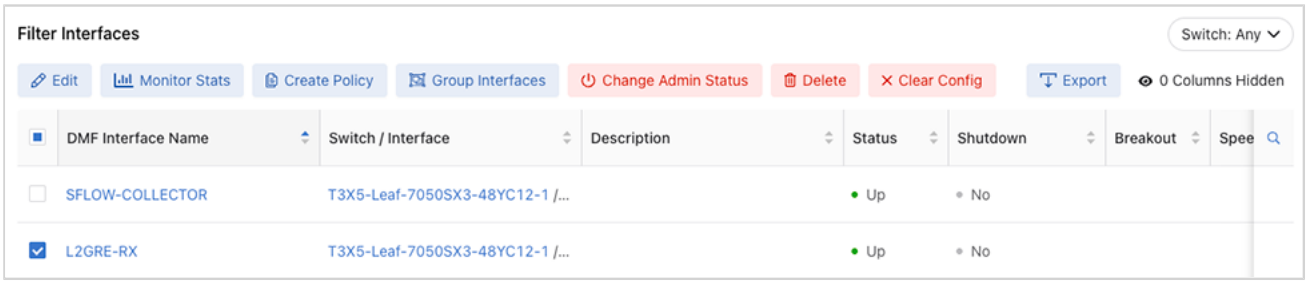


4. Select **Save** to continue.

Edit Interface

5. Select the row menu of the filter interface to configure or edit, and select **Edit**.

Figure 2-77: DANZ Monitoring Fabric (DMF) Interfaces - Edit

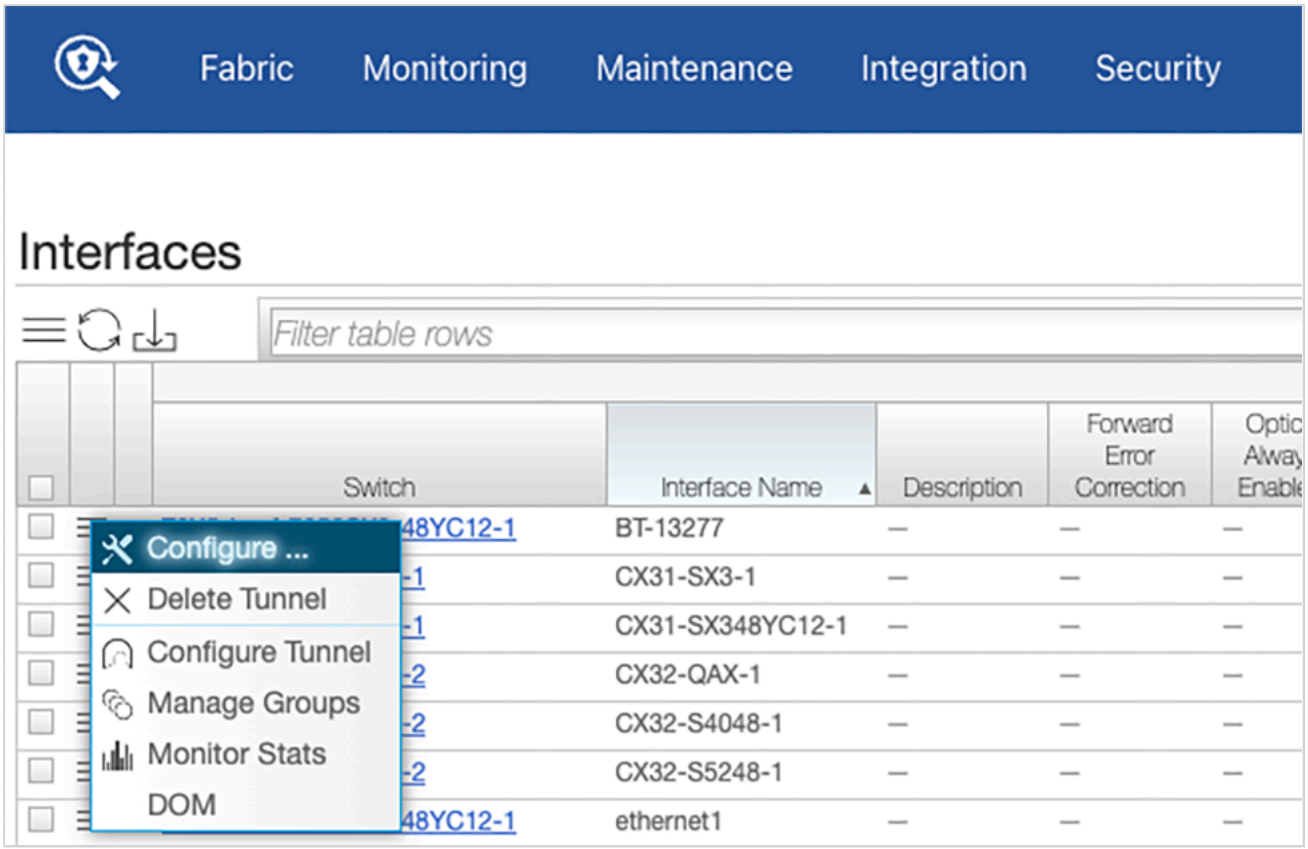


Workflow Two

Navigate to **Fabric** > **Interfaces** and perform the following steps.

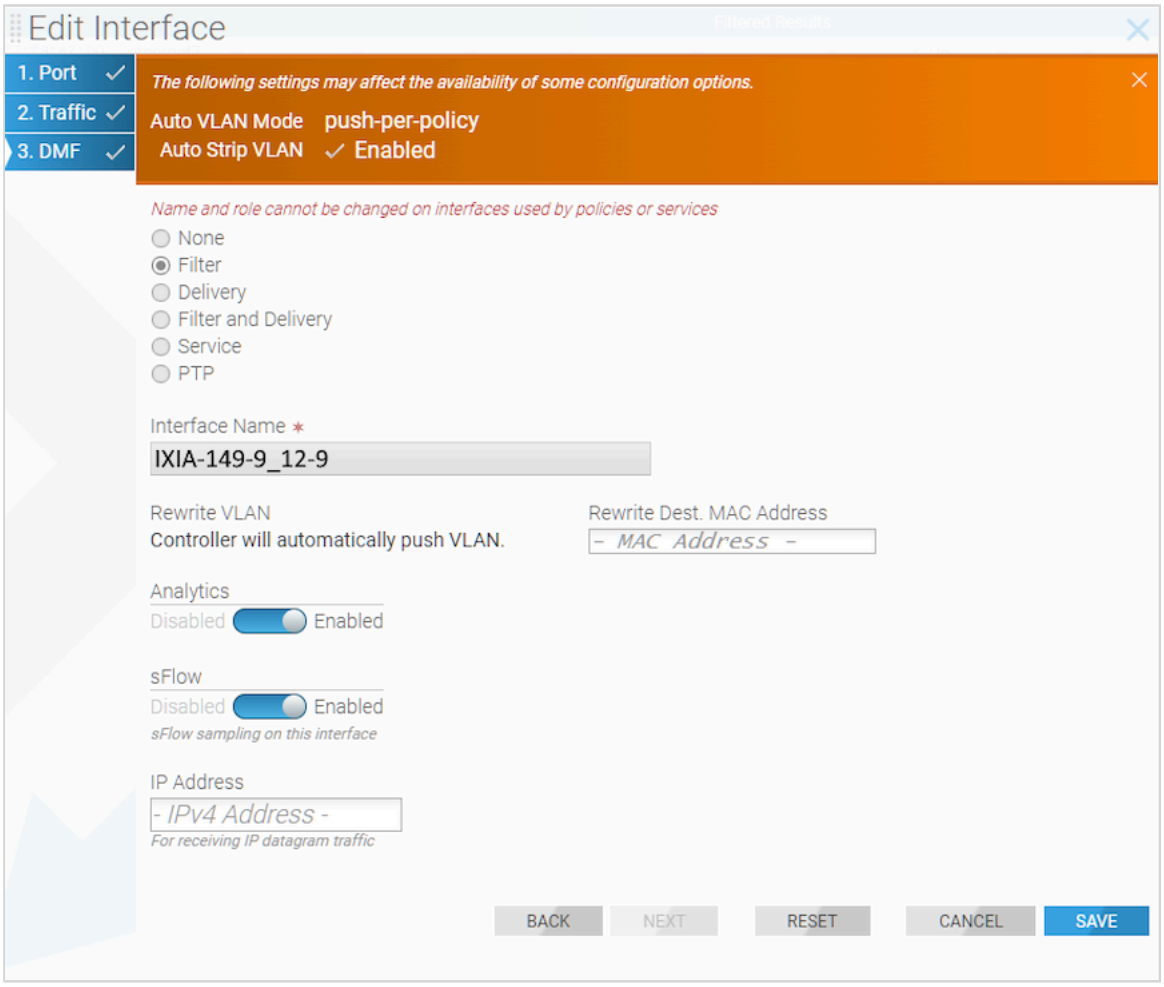
6. In the **Configure** step, use the **Rewrite Dest. MAC Address** input to configure the MAC address to override.
7. Select the row menu of the switch interface associated with the filter interface and select ☰ followed by **Configure**.

Figure 2-78: Configure Interface



8. In the **DMF** tab, select the **Rewrite Dest.MAC Address** field to enter the MAC address to be overridden.

Figure 2-79: Edit Interface DMF Rewrite Dest. MAC Address



9. Select **Save** to continue.

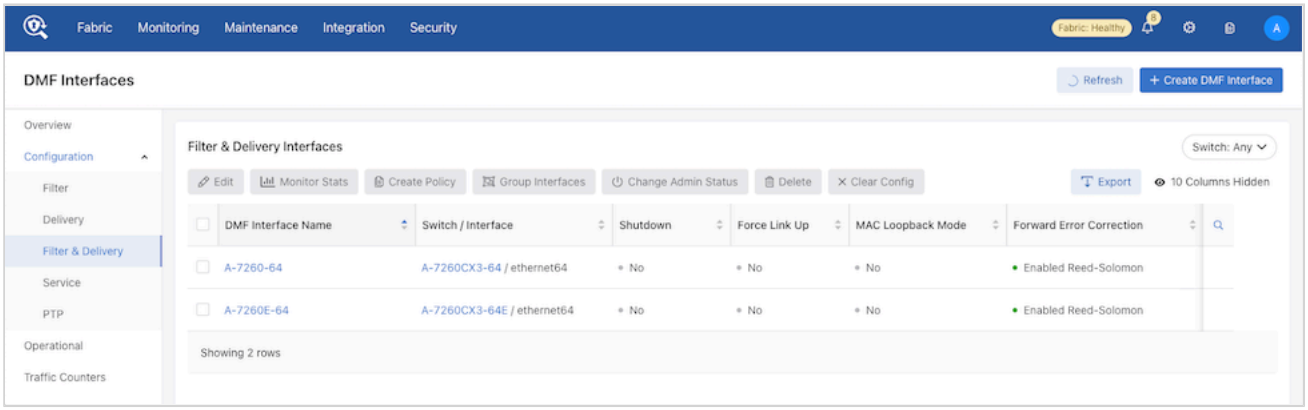
2.3.7 Configure sFlow® on a Filter and Delivery Interface

A DANZ Monitoring Fabric (DMF) interface used by a DMF policy as both a filter and a delivery interface is known as a filter-and-delivery interface. Filter-and-delivery interfaces support configuring sFlow®* in the DMF Controller. DMF enables sFlow by default on filter-and-delivery interfaces.

To disable sFlow on a filter-and-delivery interface, use the DMF Controller GUI.

1. Navigate to **Monitoring** > **Interfaces** > **Filter & Delivery**.

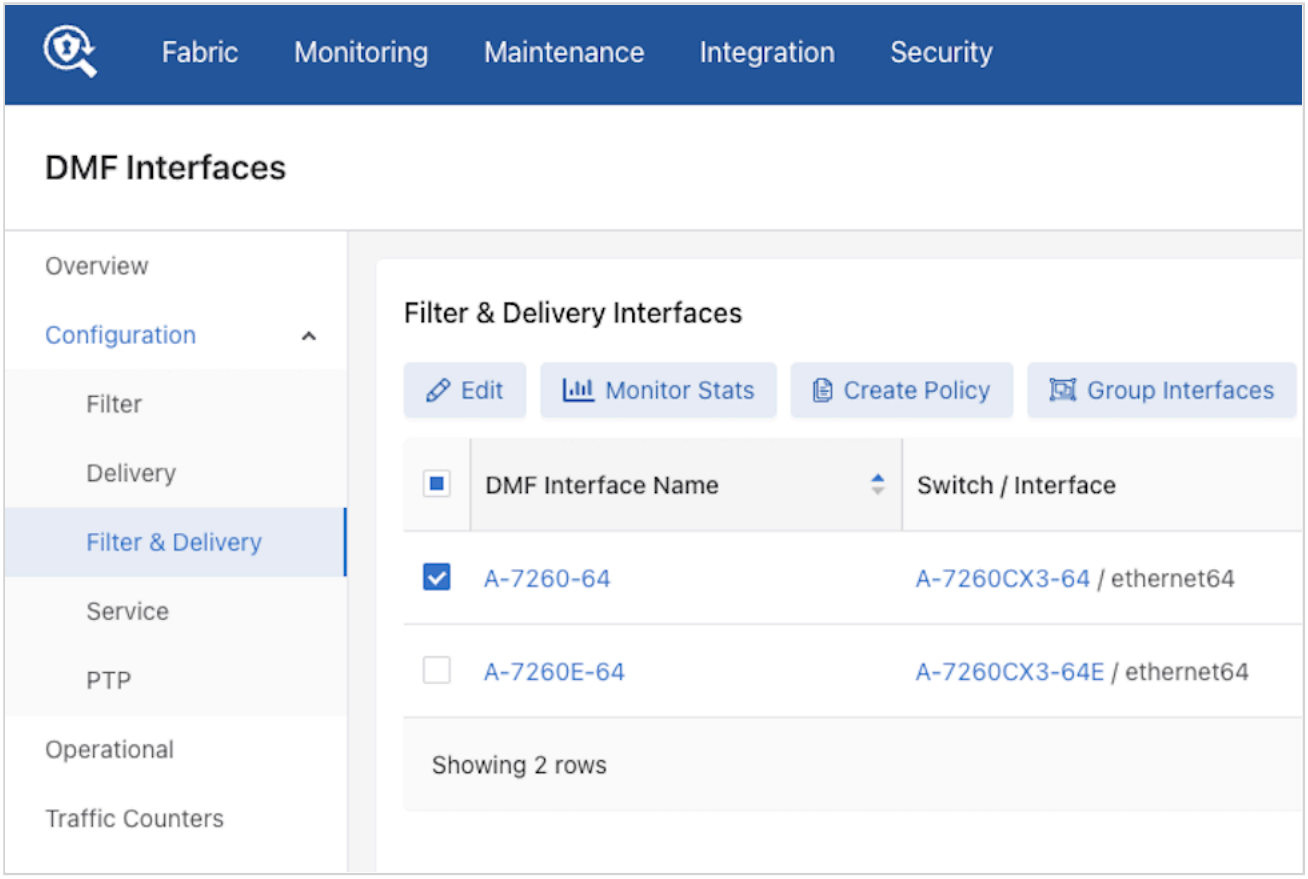
   Figure 2-80: DMF Interfaces



* sFlow® is a registered trademark of Inmon Corp.

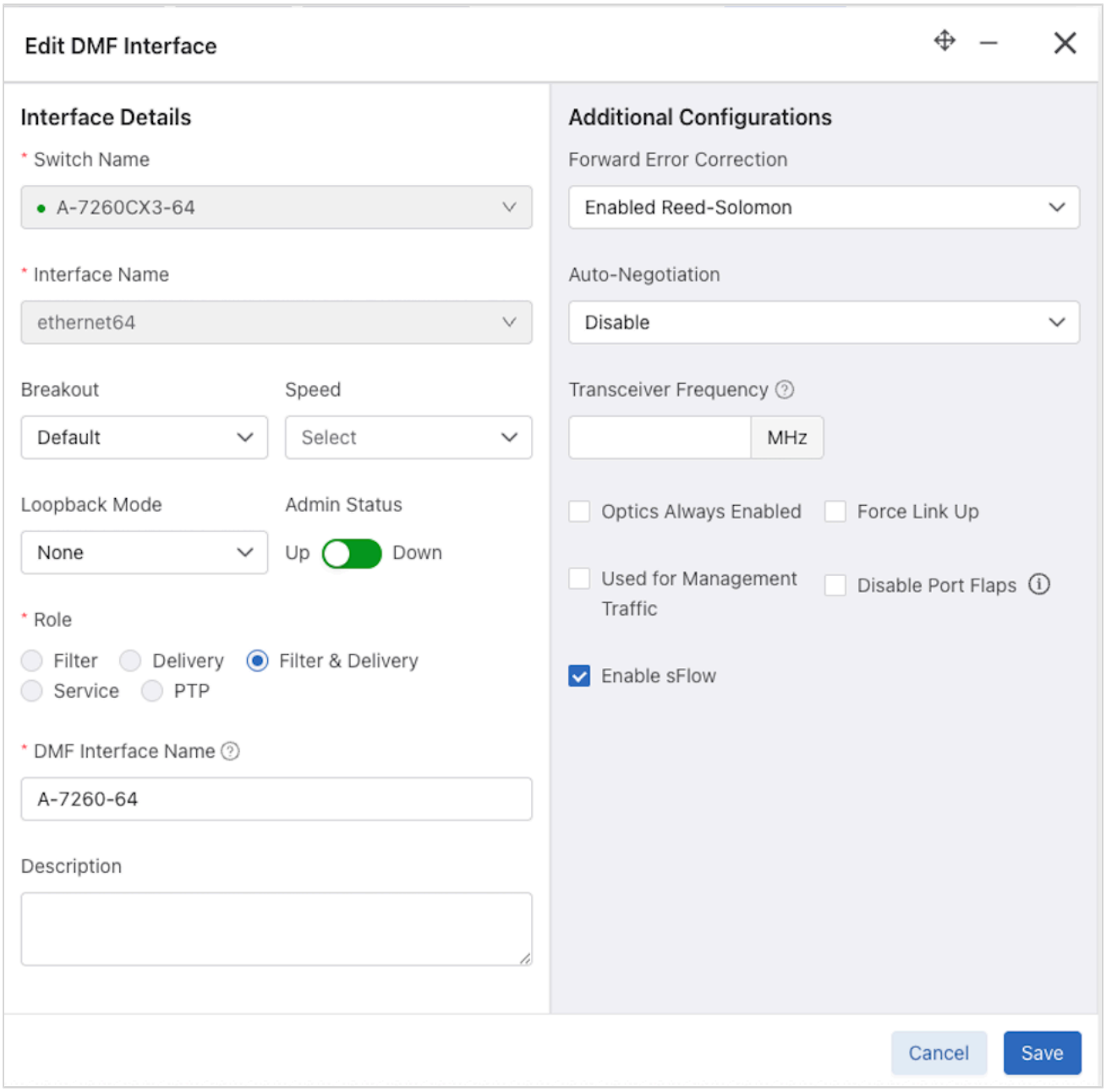
2. Select an interface from the list under **DMF Interface Name**.

Figure 2-81: DMF Interface Name



3. Select **Edit** to open the **Edit DMF Interface** configuration window.

Figure 2-82: Edit DMF Interface



4. Clear the **Enable sFlow** checkbox and select **Save**.

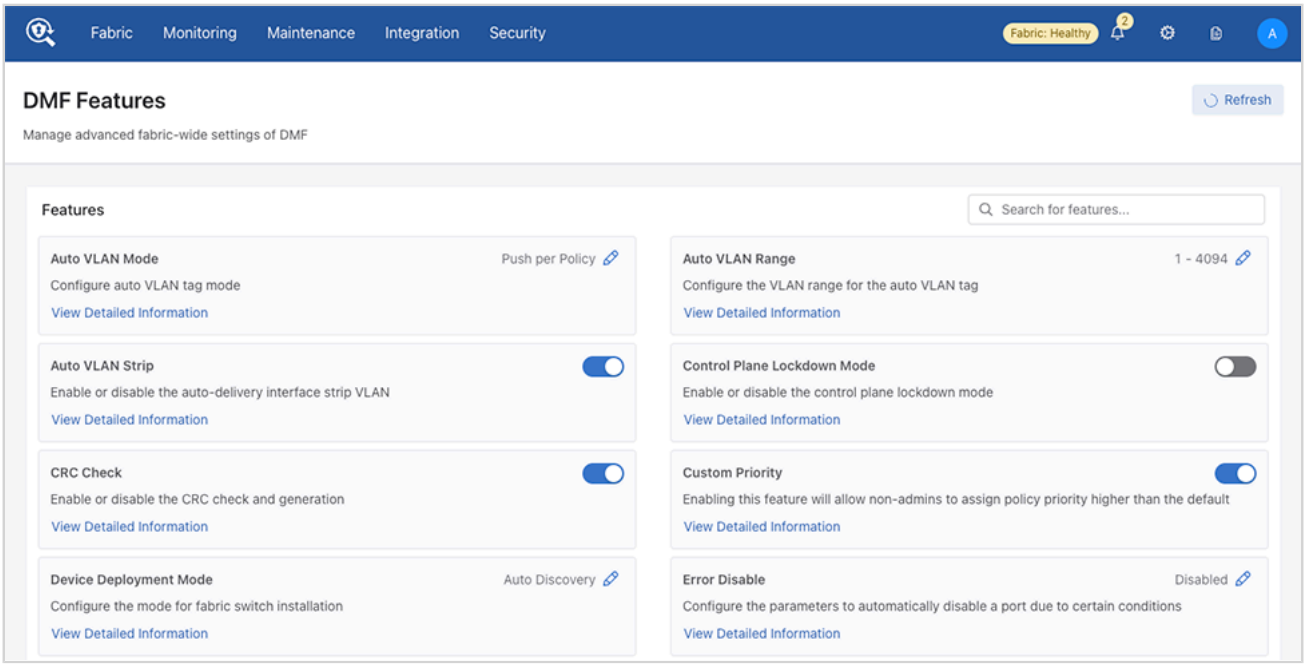
2.3.8 Configure Error Disable State

Error Disable State for Port Flapping prevents policy churn by automatically placing switch interfaces with frequent flapping into an error-disabled state, effectively performing an automatic administrative shutdown. The feature allows for automatically recovering these interfaces after a specified time. It reduces the risk of lost packets caused by continuous re-computation of DANZ Monitoring Fabric (DMF) policies due to flapping interfaces. By default, Error Disable State for Port Flapping is disabled.

Global Configuration

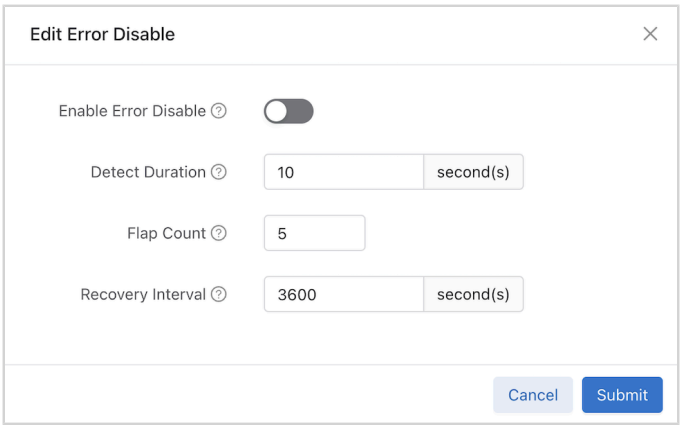
Navigate to **DMF Features** > **Error Disable** using the **gear** icon.

Figure 2-83: DMF Features



Locate the **Error Disable** feature card and select the **pencil** (edit) icon.

Select **Enable Error Disable** to enable or disable the feature.



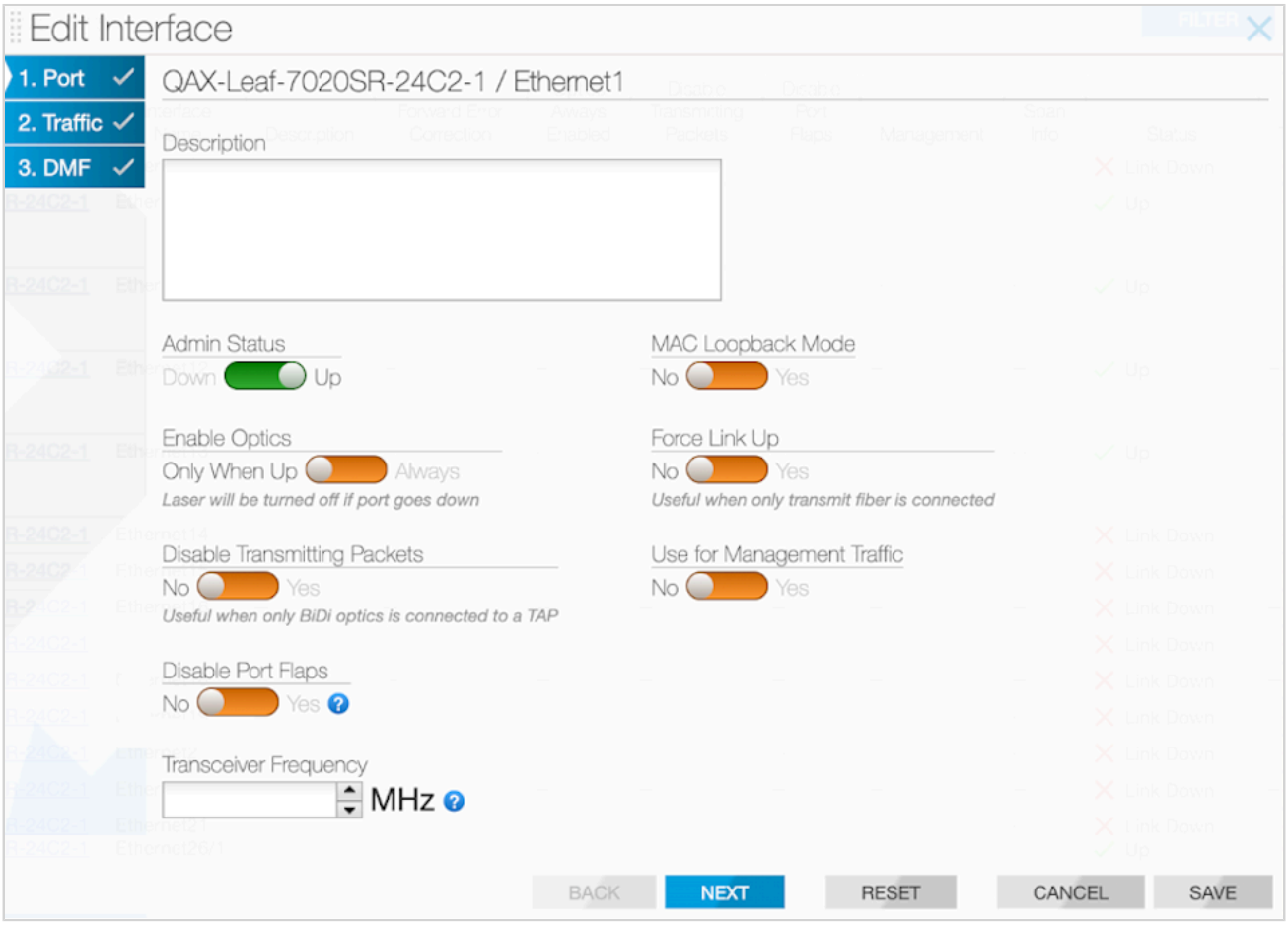
Configure the following inputs as required:

- **Enable Error Disable**: Enable or disable the feature.
- **Detect Duration**: Specifies the duration during which the required flap count must be reached to set the interface into an error-disabled state. The default is **10** seconds.
- **Flap Count**: Specifies the number of flaps required within the detect-duration period to set the interface into the error-disabled state. The default is **5** flaps.
- **Recovery Interval**: The **Recovery Interval** indicates how much time will pass before bringing up the interface when the automatic administrative shutdown action shuts it down. The default is **3600** seconds.

Select **Submit** to save the global configuration.

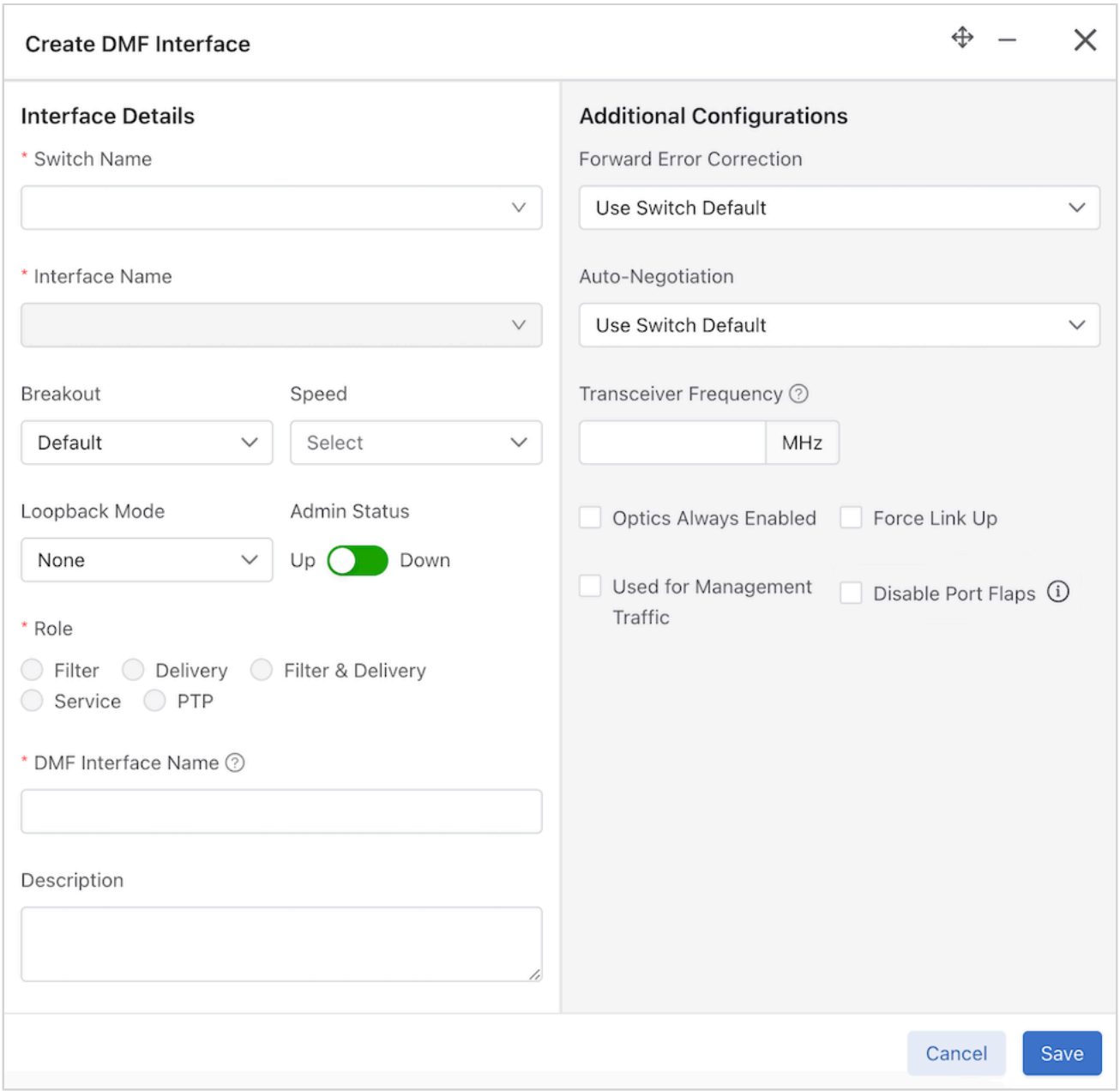
Configure Error Disable at the Interface Level

Use the **Disable Port Flaps** switch to enable or disable port flaps at the interface level on the **Fabric** > **Interfaces** dashboard.



Select **Save**.

When using the **Monitoring** > **Interfaces** dashboard during an **Edit Interface** step or **Create DMF Interface** configuration process use the **Disable Port Flaps** checkbox to enable or disable port flaps on the **Create DMF Interface** page. If the **Error Disable** feature is enabled globally, checking this input will disable the **Error Disable** feature on this interface temporarily.



Select **Save**.

Error Disable Alerts

Select the **Alarm** icon followed by **Warnings** to view any Error Disable warnings. These appear on the **Alerts** page.

Figure 2-84: Warnings

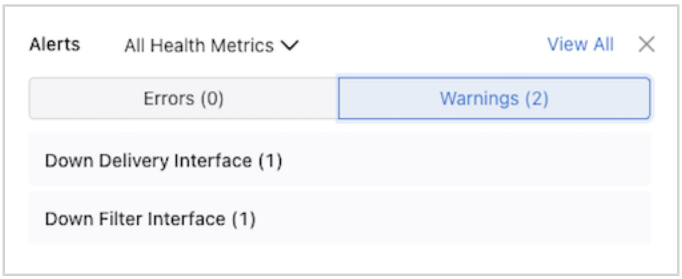
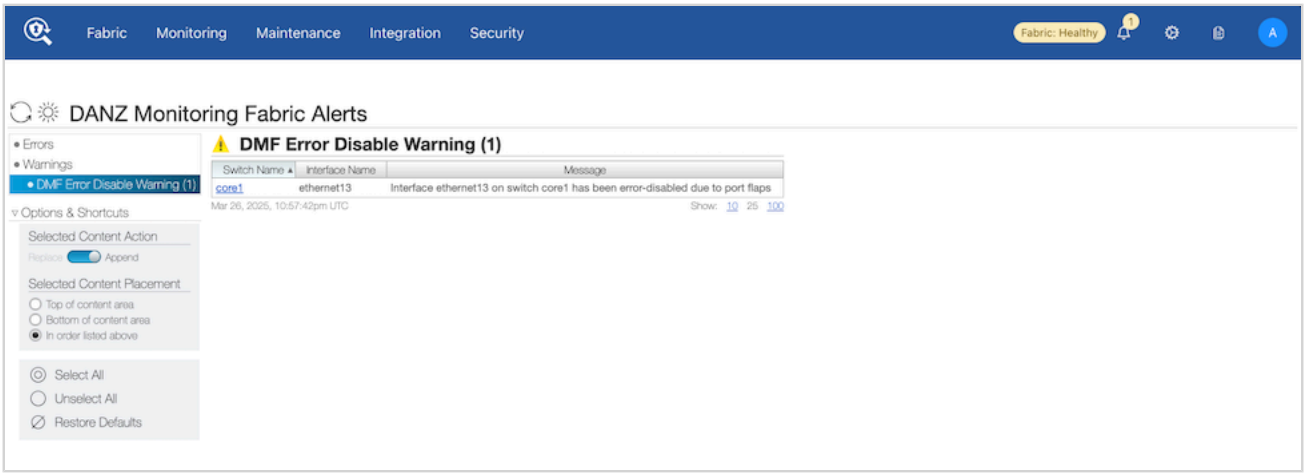


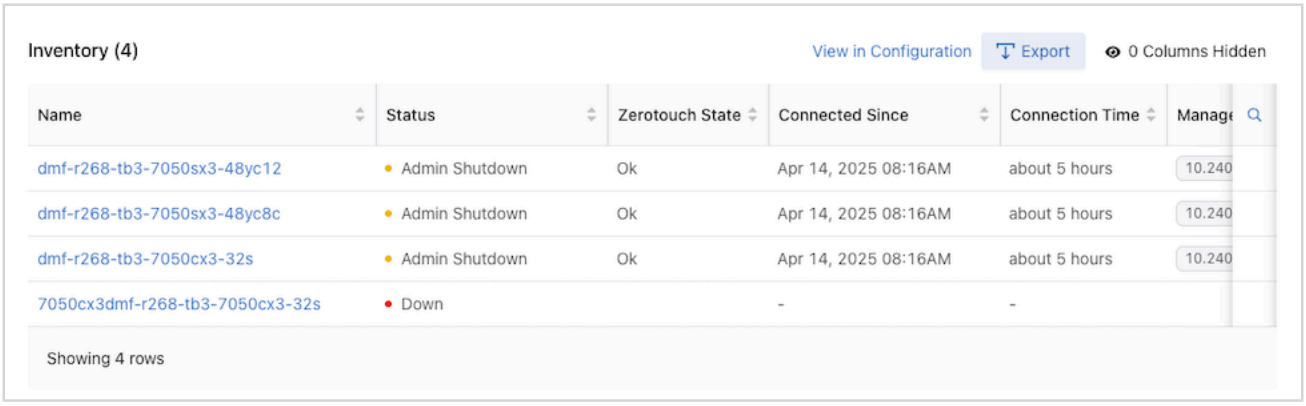
Figure 2-85: DMF Error Disable Warnings



Details related to error disable appear in the **Switches** detail page in the **Error Disable** dashboard.

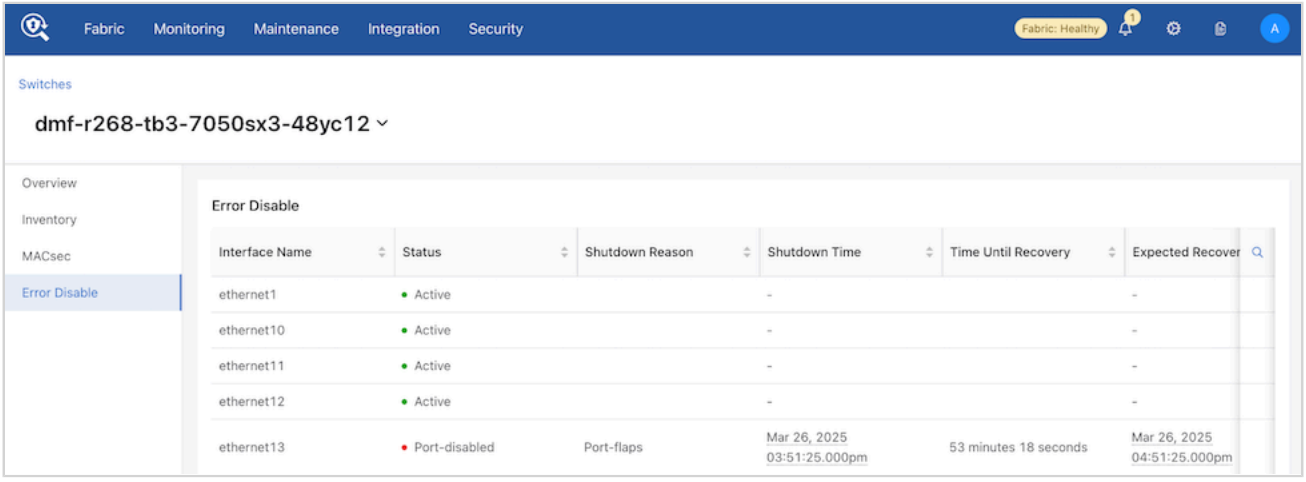
Navigate to **Fabric** > **Switches** > **Inventory** and choose a switch from the inventory list.

Figure 2-86: Inventory



Select **Error Disable**.

Figure 2-87: Error Disable Dashboard



2.3.9 Configure Interface Groups

DMF version 8.8.0 introduces a redesigned workflow for Interface Groups in the DMF UI. An interface group is a collection of one or more filter or delivery interfaces, making it more convenient to create a policy. It is often easier to refer to an interface group when creating a policy than to identify every interface to which the policy applies explicitly.

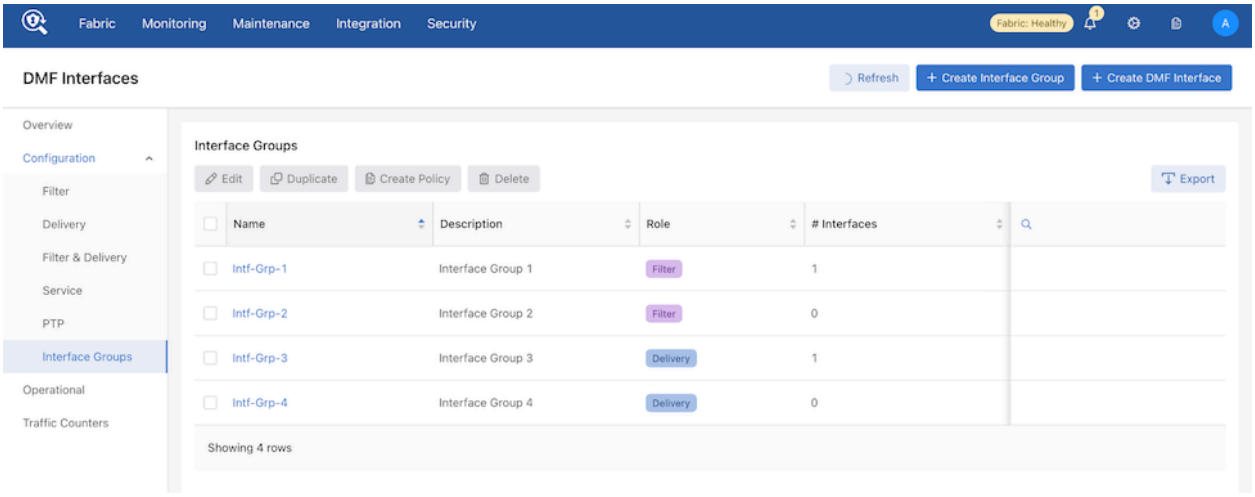
Create an interface group consisting of one or more filter or delivery interfaces. Use an address group in multiple policies, referring to the IP address group by name in match rules. If no subnet mask is provided in the address group, it is assumed to be an exact match. For example, in an IPv4 address group, the absence of a mask implies a mask of /32. For an IPv6 address group, the absence of a mask implies a mask of /128. Identify only a single IP address group for a specific policy match rule. Address lists with both src-ip and dst-ip options cannot exist in the same match rule.

Important: An Interface Group differs from a Link Aggregation Group (LAG). A LAG combines multiple LAN links and cables in parallel to provide a high level of redundancy and increased transmission speed.

To access the **Interface Groups** workflow, and perform the following steps:

1. Navigate to **Monitoring** > **Interfaces**.

Figure 2-88: DMF Interfaces - Interface Groups



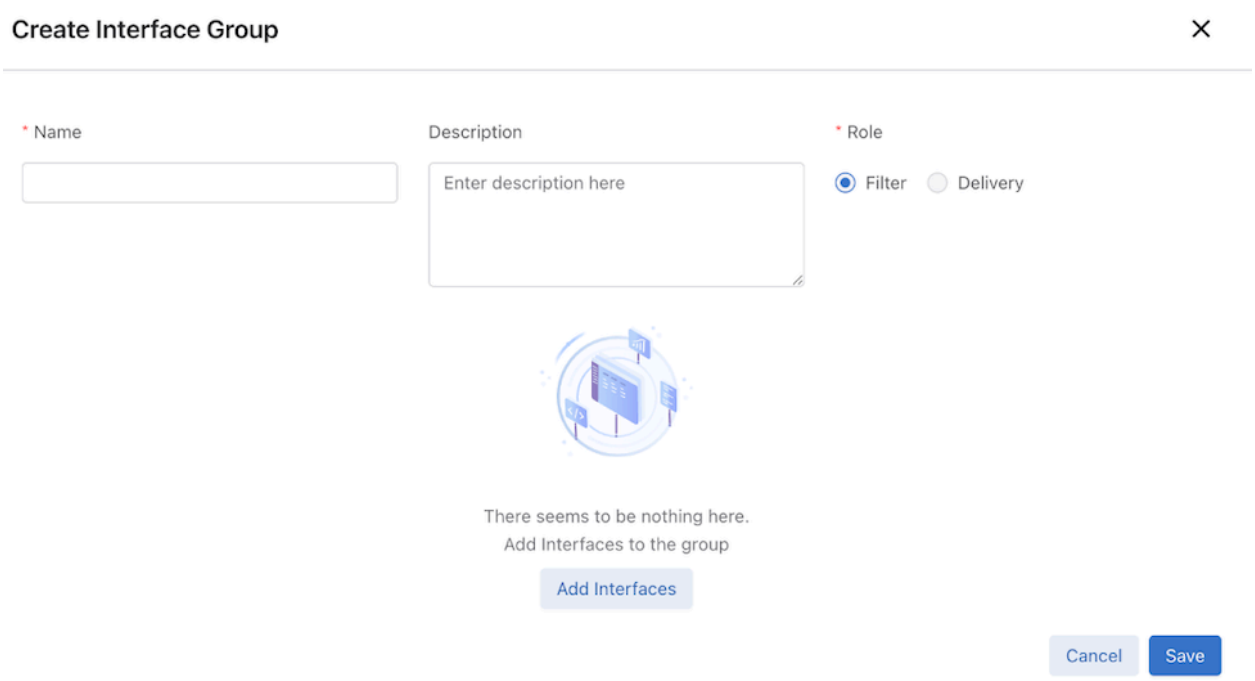
A **+ Create Interface Group** control displays to the left of **Create DMF Interface**.

Figure 2-89: + Create Interface Group



2. Select **+ Create Interface Group**.
The **Create Interface Group** workflow displays:

Figure 2-90: Creating Interface Group



3. Select **Add Interfaces** to display a list of available interfaces according to the selected **Role**, either **Filter** or **Delivery**.
   - If **Filter** is selected, DMF displays interfaces with **Filter** and **Filter & Delivery** roles.
   - If **Delivery** is selected, DMF displays interfaces with **Delivery** and **Filter & Delivery** roles.

   These options ensure all compatible interfaces are available for selection.

Figure 2-91: Add Filter Interfaces

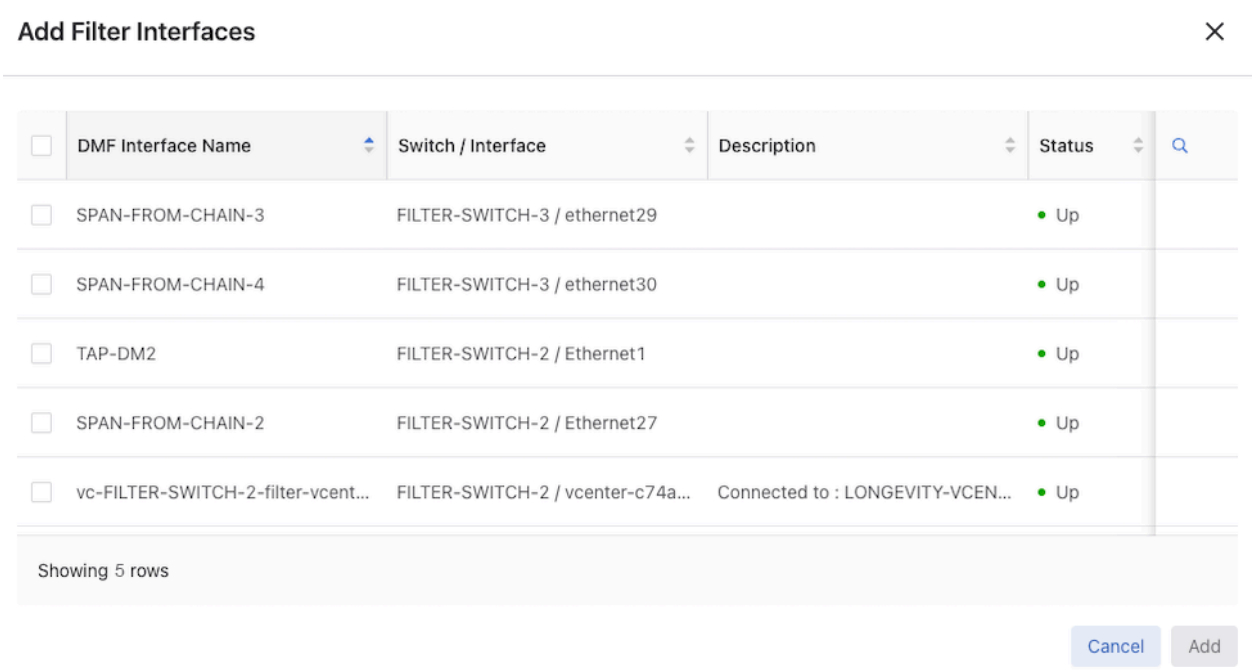
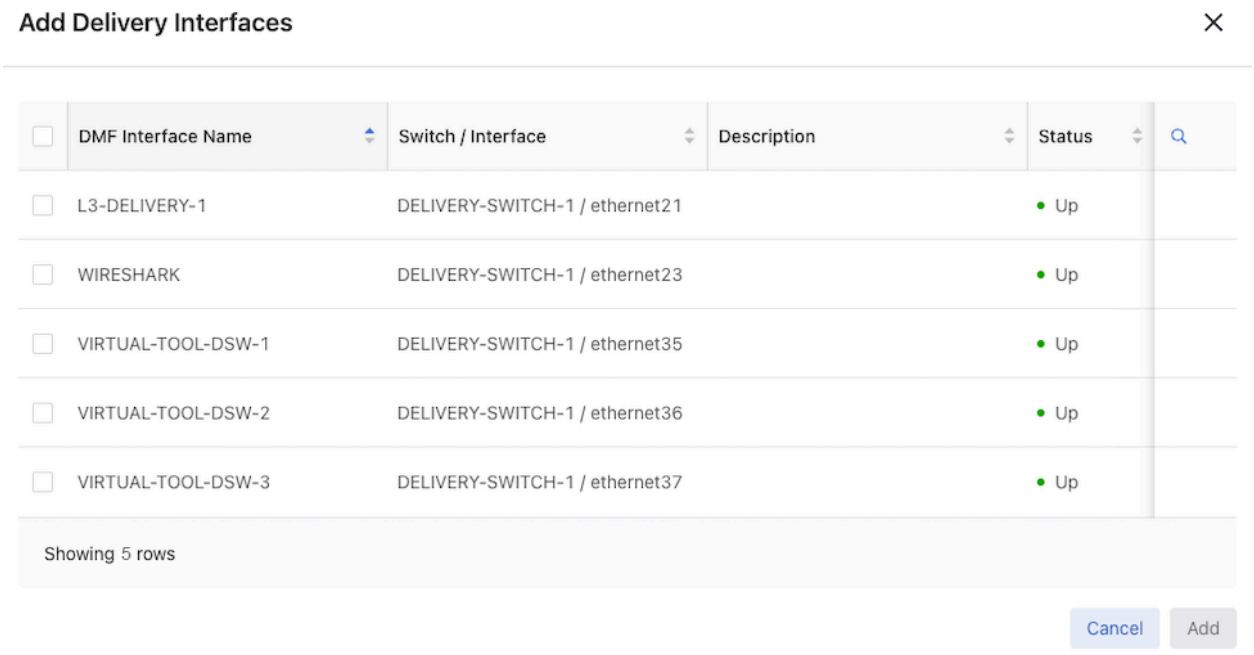
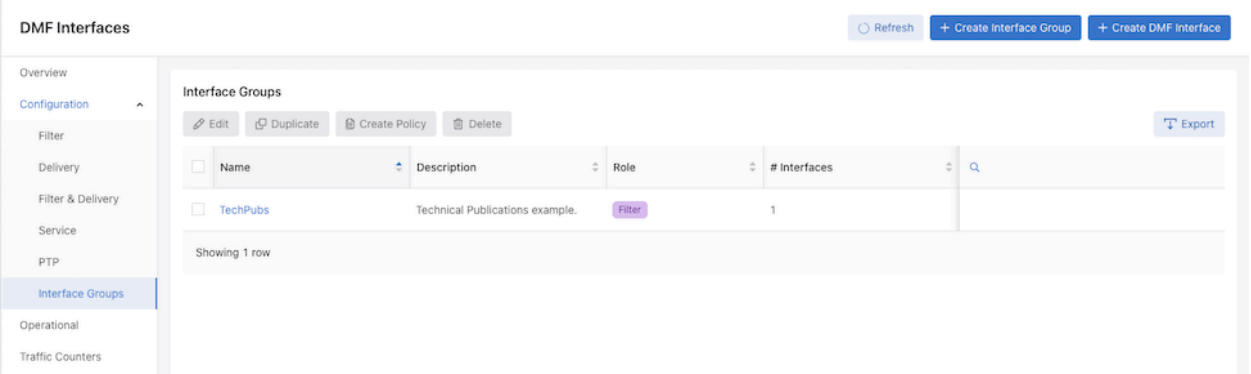


Figure 2-92: Add Delivery Interfaces



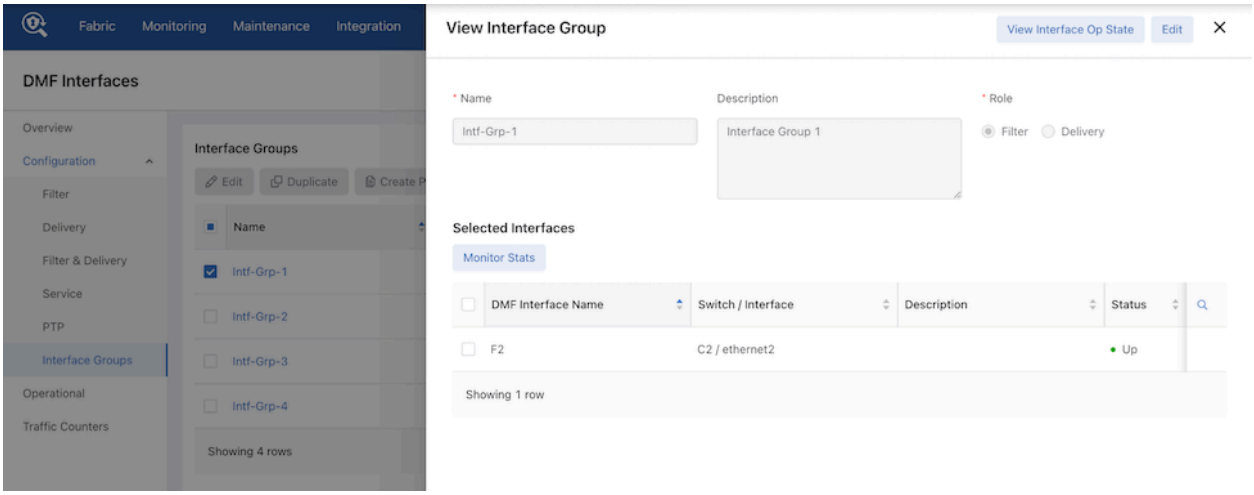
4. Select **Add** to add the interfaces to the group.
5. Enter the information for the **Interface Group** in the following fields:
   - **Name**: Required - unique identifier.
   - **Description**: Optional - text field.
   - **Role**: Required - as described in Step 3.
6. Select **Save** to complete the configuration.
   The new group displays in the **Interface Groups** dashboard.

Figure 2-93: New Group



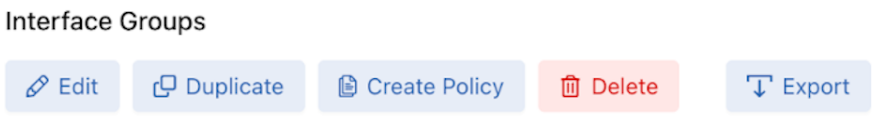
7. View details for each **Interface Group** by selecting the corresponding group name in the table.
   The following action buttons are available within the **Interface Group** details view:
   - **Edit**: Enables edit mode for the selected Interface Group.
   - **View Interface Op State**: Redirects to the Operational section with the relevant Interface Group pre-selected.
   - **Monitor Stats**: This option allows the selection of one or more interfaces. When interfaces are selected, the Traffic Counters page opens with those pre-selected interfaces. If no interfaces are selected, the page opens with default interface selections.

Figure 2-94: View Interface Group



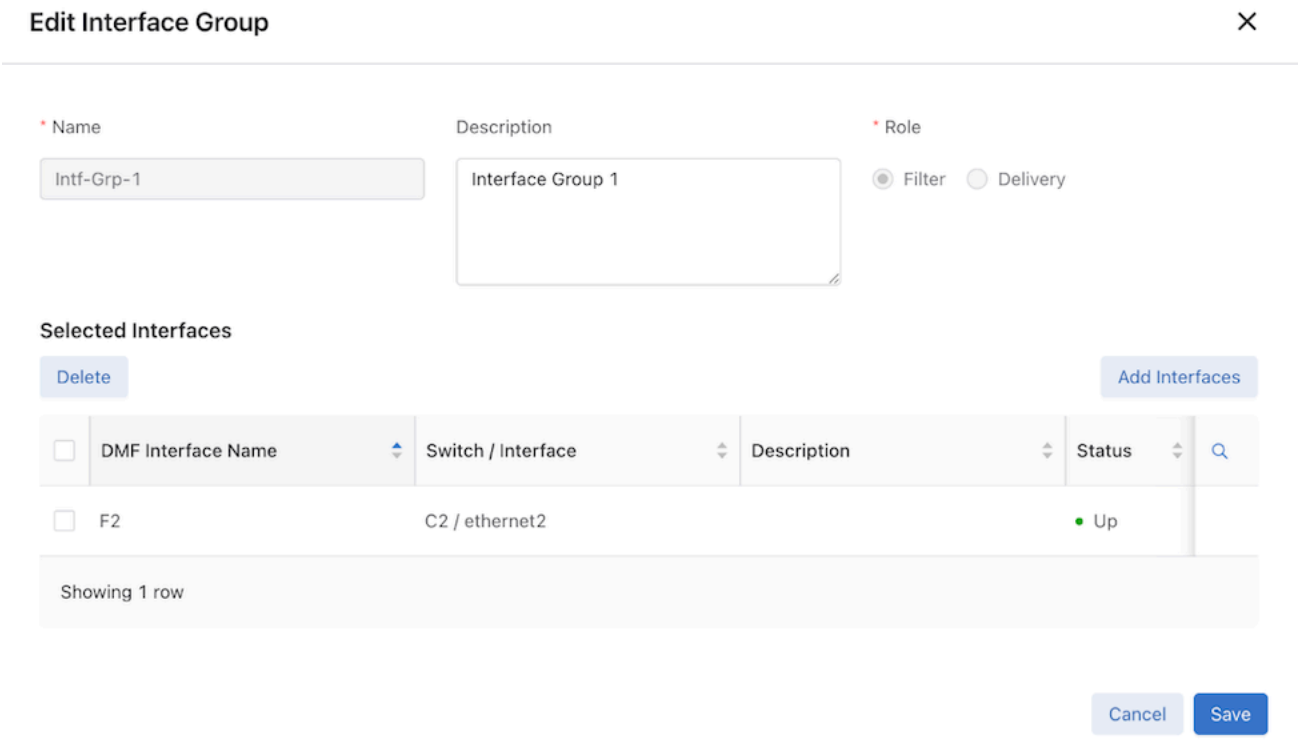
Additional Interface Controls

Action Buttons



Edit - Select a single row from the list to edit an interface group.

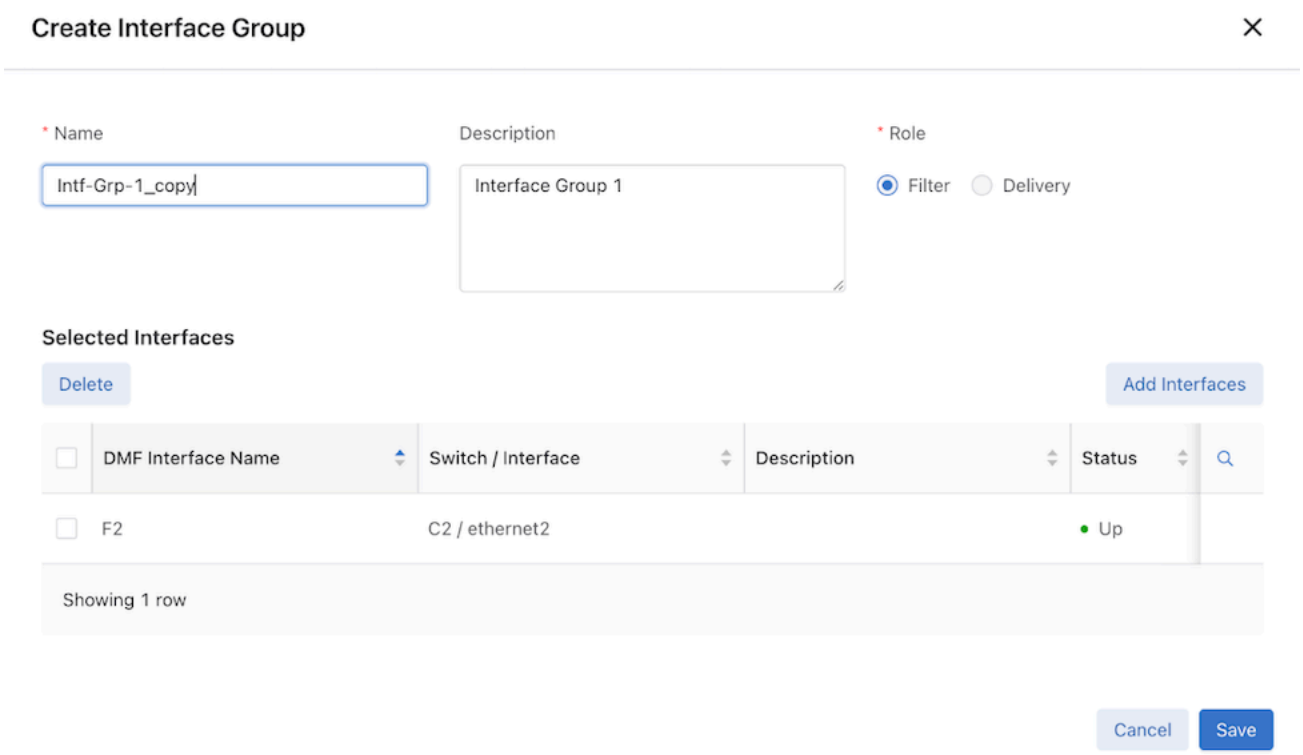
Figure 2-95: Edit Interface Group



Note: Only the **Description** and **Interfaces** fields are editable. The **Name** and **Role** fields are read-only and cannot be modified.

Duplicate - To duplicate an interface group, select a single row from the list that opens a copy of the selected interface group for editing.

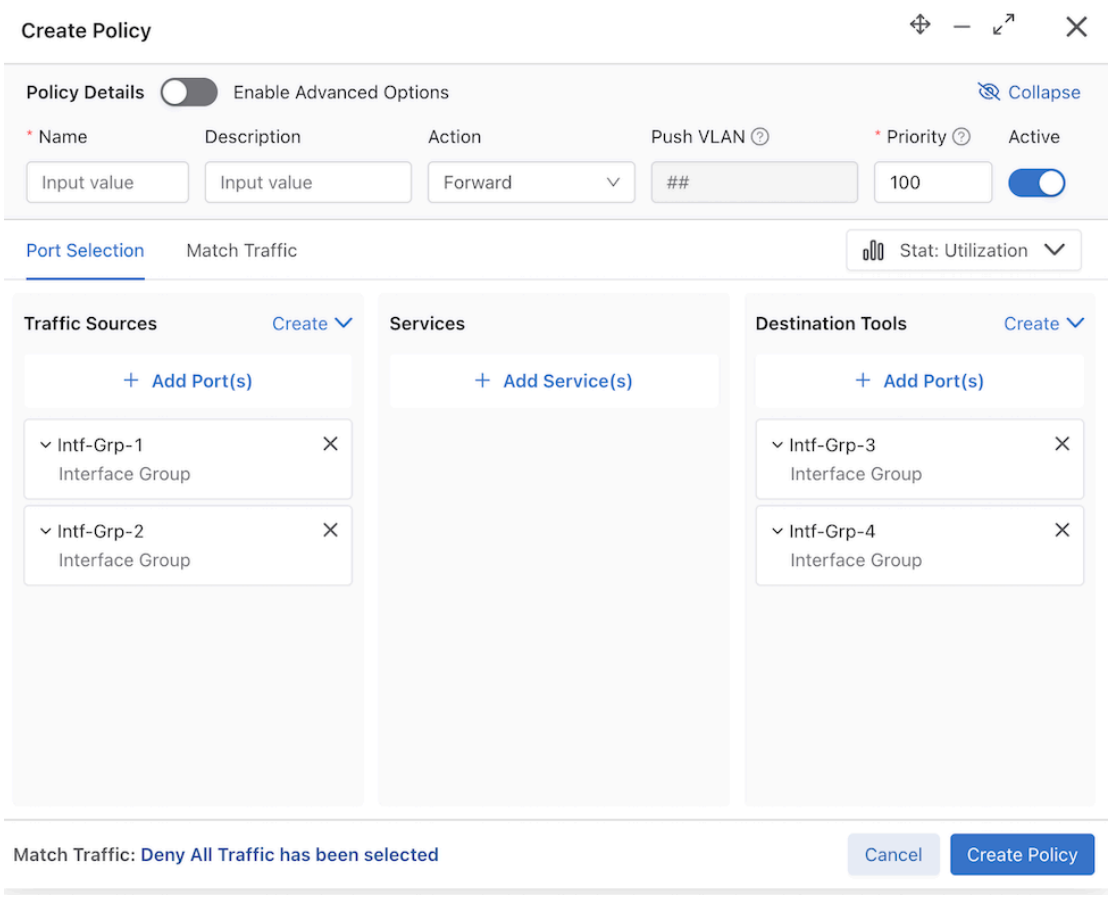
Figure 2-96: Duplicate Interface



Note: All fields in the duplicated group are editable. The **Name** field is automatically updated by appending "_copy" to the original name.

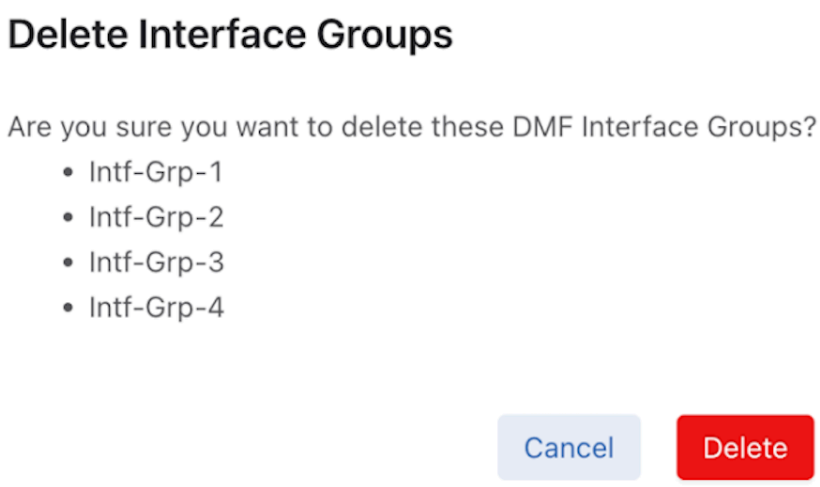
Create Policy - The **Create Policy** option allows the creation of policies directly from the **Interface Groups** page. Upon selecting **Create Policy**, the selected Interface Groups are automatically organized under different sources based on their **Role** — either **Filter** or **Delivery**.

Figure 2-97: Create Policy



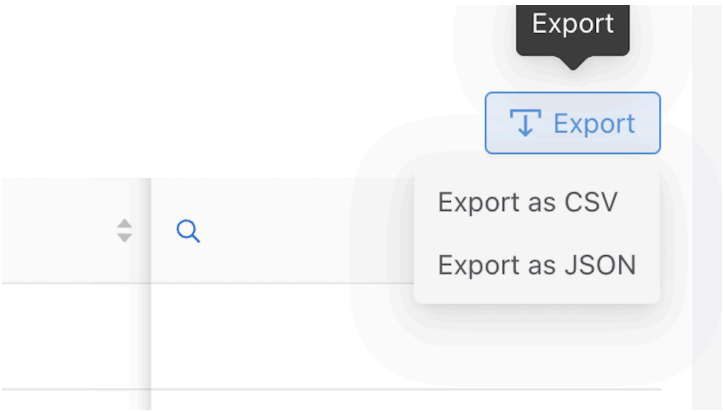
Delete - Delete interface groups by selecting one or more groups and using the **Delete** option.

Figure 2-98: Delete Interface Group



Export - The **Export** option for the Interface Groups table supports both **CSV** and **JSON** formats. Export functionality is independent of any row selection and includes the complete table data.

Figure 2-99: Export Data



2.3.9.1 Considerations

During Interface Group creation, interface selections are not persistent when switching between roles. When adding interfaces to one role and switching to another, the selections made for the first role are lost.

Example:

1. Select the role **Filter** and add interfaces.
2. Change the role to **Delivery**.
3. Switch back to the **Filter** role — the previously selected interfaces are gone.

When deleting a previously added interface in an Interface Group, DMF does not **automatically remove** it from the Interface Group. The deleted interface entry must be **manually removed** from the Interface Group configuration.

2.3.10 Using the Command Line Interface

Configure a Filter or Delivery Interface

Interface Description

Identify a Filter Interface using Destination MAC Rewrite

Forward Slashes in Interface Names

Configure sFlow on a Filter and Delivery Interface

Configure Interface Groups

Configure Error Disable State

Shows Commands - Error State

Troubleshooting - Error State

Overriding the Default Configuration for a Switch

2.3.10.1 Configure a Filter or Delivery Interface

To assign a filter or delivery role to an interface, perform the following steps:

1. From the config mode, enter the **`switch`** command, identifying the switch having the interface to configure.

```
controller-1(config)# switch DMF-FILTER-SWITCH-1
controller-1(config-switch)#
```



Note: Identify the switch using the alias if configured. The CLI changes to the config-switch submode to configure the specified switch.

2. From the config-switch mode, enter the **`interface`** command, as in the following example:

```
controller-1(config-switch)# interface ethernet1
controller-1(config-switch-if)#
```



Note: To view a list of the available interfaces, enter the **`show switch`** ***switch-name interface*** command or press the **Tab** key, and the command completion feature displays a concise list of permitted values. After identifying the interface, the CLI changes to the config-switch-if mode to configure the specified interface.

3. From the **`config-switch-if`** submode, enter the **`role`** command to identify the role for the interface. The syntax for defining an interface role (**delivery**, **filter**, **filter-and-delivery**, or **service**) is as follows:

```
[no] role delivery interface-name <name> [strip-customer-vlan] [ip-address
<ip-address>]
[nexthop-ip <ip-address> <subnet> ]
[no] role filter interface-name <name> [ip-address <ip-address>] {[rewrite
vlan <vlan id (1-4094)>]} [no-analytics]
[no] role both-filter-and-delivery interface-name <name> {[rewrite vlan <vlan id
(1-4094)>]} [noanalytics]
[no] role service interface-name <name>a
```

The interface-name command assigns an alias to the current interface, which typically would indicate the role assigned, as in the following example:

```
controller-1(config-switch-if)# role delivery interface-name TOOL-PORT-1
```



Note: An interface can have only one role, and the configured interface name must be unique within the DANZ Monitoring Fabric.

The following examples show the configuration for filter, delivery, and service interfaces:

- **Filter Interfaces**

```
controller-1 (config)# switch DMF-FILTER-SWITCH-1
controller-1(config-switch)# interface ethernet1
controller-1(config-switch-if)# role filter interface-name TAP-PORT-1
controller-1(config-switch-if)# interface ethernet2
controller-1(config-switch-if)# role filter interface-name TAP-PORT-2
```

- **Delivery Interfaces**

```
controller-1(config-switch-if)# switch DMF-DELIVERY-SWITCH-1
controller-1(config-switch-if)# interface ethernet1
controller-1(config-switch-if)# role delivery interface-name TOOL-PORT-1
controller-1(config-switch-if)# interface ethernet2
controller-1(config-switch-if)# role delivery interface-name TOOL-PORT-2
```

- **Filter and Delivery Interfaces**

```
controller-1(config-switch-if)# switch DMF-CORE-SWITCH-1
controller-1(config-switch-if)# interface ethernet1
controller-1(config-switch-if)# role both-filter-and-delivery interface-name loopback-
port-1
controller-1(config-switch-if)# interface ethernet2
controller-1(config-switch-if)# role both-filter-and-delivery interface-name loopback-
port-2
```

- **Service Interfaces**

```
controller-1(config-switch-if)# switch DMF-CORE-SWITCH-1
controller-1(config-switch-if)# interface ethernet1
controller-1(config-switch-if)# role service interface-name PRE-SERVICE-PORT-1
controller-1(config-switch-if)# interface ethernet2
controller-1(config-switch-if)# role service interface-name POST-SERVICE-PORT-1
```

Note:

a. An interface can have only one role, and the configured interface name must be unique within the DANZ Monitoring Fabric.

b. A delivery interface will show drops under a many-to-one scenario, i.e., multiple filter interfaces pointing to a single delivery interface as per policy definition. These drops are accounted for as micro bursts occur at the egress port. For example, consider a use case of three 10G ingress ports and one 25G egress port. Even if we send a total of 25Gbps of traffic by calculation from ingress to egress, each individual ingress port still operates at 10Gbps inside the BCM chip (i.e., a total of 30G on ingress, 5Gbps traffic is still running at 10Gbps speed on the wire but with a bigger inter-frame gap). This means the ingress may oversubscribe the egress due to the 30G to 25G traffic ratio. For example, if each ingress port receives one packet at the same time, it causes 30G-to-25G over-subscription or micro-bursting (5Gbps traffic still gets processed at the ingress port's native speed of 10Gbps). Because the egress can only process packets up to 25Gbps, one of the packets will not get dequeued promptly and accumulate inside the egress TX queue. If this pattern continues, the egress queue eventually drops packets due to the TX buffer becoming full. Therefore, expect this behavior in the case of many-to-one forwarding. After reconfiguring and using only one 25G ingress port to one 25G egress port, there is no TX drop problem.

2.3.10.2 Interface Description

The **show switch <switch name/all> interface details** or **show switch <switch name/all> interface <interface name> details** commands in the CLI include a **Description** column, which provides the configured description (if any) for the corresponding interface. This is a only a CLI change.

No configuration is necessary to use this feature. However, the column remains empty for any switch interface without a configured description. Please refer to the existing user documentation to configure a description for an interface.

Show Commands

Use the following **show** commands for viewing an interface description.

Display the interface **Description** in **show switch *name* interface** command:

```
dmf-controller# show switch all interface details
#   Switch    IF Name    MAC Address                  Config State Adv. Features Curr Features
 Supported Features Description
---|---------|----------|----------------------------|------|-----|-------------|--------
-----|-----------------|-----------------------------------|
1   core1     ethernet1  52:54:00:d9:58:8c (Linux KVM) up     up    10g           10g
 10g
2   core1     ethernet2  52:54:00:5a:ba:93 (Linux KVM) up     up    10g           10g
 10g               sample description for eth2
...
```

With the switch name provided:

```
dmf-controller# show switch core1 interface details
#  IF Name    MAC Address                  Config State Adv. Features Curr Features Supported
 Features Description
--|----------|----------------------------|------|-----|-------------|-------------|-----
-------------|--------------------------------|
1  ethernet1  52:54:00:d9:58:8c (Linux KVM) up     up    10g           10g           10g
2  ethernet2  52:54:00:5a:ba:93 (Linux KVM) up     up    10g           10g           10g
      sample description for eth2
3  ethernet3  5c:16:c7:10:e0:bc (Arista)    up     down  10g           10g           10g
4  ethernet4  5c:16:c7:10:e0:bd (Arista)    up     down  10g           10g           10g
5  ethernet5  5c:16:c7:10:e0:be (Arista)    up     down  10g           10g           10g
      sample description for eth5
...
```

With the switch name and the interface name provided:

```
dmf-controller# show switch core1 interface ethernet5 details
# IF Name   MAC Address               Config State Adv. Features Curr Features Supported Features
 Description
-|---------|--------------------------|------|-----|-------------|-------------|----------
--------|--------------------------------|
1 ethernet5 5c:16:c7:10:e0:be (Arista) up     down  10g           10g           10g
 sample description for eth5
```

2.3.10.3 Identify a Filter Interface using Destination MAC Rewrite

The Destination MAC (D.MAC) Rewrite feature provides an option to identify the Filter interface by overriding the destination MAC address of the packet received on the filter interface. Use this feature for auto-assigned and user-configured VLANs in push-per-filter and push-per-policy modes.



Note: The D.MAC Rewrite feature VLAN preservation applies to switches running SWL OS and does not apply to 7280R/7280R2 switches running EOS.

Global Configuration

Configure this function at the filter interface level and perform the following steps using the CLI.

1. Select a filter switch and enter the config mode.

```
(config)# switch filter1
```

2. Select an interface from the switch acting as the filter interface.

```
(config-switch)# interface ethernet5
```

3. Create a filter interface with a name and provide the MAC address to override.

```
(config-switch-if)# role filter interface-name f1 rewrite dst-mac 00:00:00:00:00:03
```

CLI Show Commands

The following **`show`** command displays the ingress flow for the filter switch.

In the `Entry value` column, the filter switch contains `dst MAC tlv: EthDst(00:00:00:00:00:03)`.

```
(config-policy)# show switch filter1 table ingress-flow-2
# Ingress-flow-2 Device name Entry key                                  Entry value
-|--------------|-----------|-------------------------------------|---------------------
--------------|
1 0              filter1     Priority(6400), Port(5), EthType(34525) Name(p1), Data([0, 0, 0,
 61]), PushVlanOnIngress(flags=[]), VlanVid(0x1), Port(1), EthDst(00:00:00:00:00:03)
2 1              filter1     Priority(6400), Port(5)                 Name(p1), Data([0, 0, 0,
 62]), PushVlanOnIngress(flags=[]), VlanVid(0x1), Port(1), EthDst(00:00:00:00:00:03)
3 2              filter1     Priority(36000), EthType(35020)         Name(__System_LLDP_Flow_),
 Data([0, 0, 0, 56]), Port(controller), QueueId(0)
```

The core and delivery switch in the `Entry value` column doesn't contain `dst MAC tlv`, as shown in the following examples.

```
(config-policy)# show switch core1 table ingress-flow-2
# Ingress-flow-2 Device name Entry key                                                  Entry value
-|--------------|-----------|-----------------------------------------------------|-------
--------------------|
1 0              core1       Priority(6400), Port(1), EthType(34525), VlanVid(0x1) Name(p1),
 Data([0, 0, 0, 60]), Port(2)
2 1              core1       Priority(6400), Port(1), VlanVid(0x1)                 Name(p1),
 Data([0, 0, 0, 59]), Port(2)
3 2              core1       Priority(36000), EthType(35020)                       Name(__System
_LLDP_Flow_), Data([0, 0, 0, 57]), Port(controller), QueueId(0)
```

```
(config-policy)# show switch delivery1 table ingress-flow-2
# Ingress-flow-2 Device name Entry key                                                    Entry value
-|--------------|-----------|-------------------------------------------------------|-------
--------------------|
1 0              delivery1   Priority(6400), Port(1), EthType(34525), VlanVid(0x1) Name(p1),
 Data([0, 0, 0, 64]), Port(6)
2 1              delivery1   Priority(6400), Port(1), VlanVid(0x1)                Name(p1),
 Data([0, 0, 0, 63]), Port(6)
3 2              delivery1   Priority(36000), EthType(35020)                      Name(__System
_LLDP_Flow_), Data([0, 0, 0, 58]), Port(controller), QueueId(0)
```

Troubleshooting

To troubleshoot the scenario where the provided destination MAC address is attached incorrectly to the filter interface. The `ingress-flow-2` table shown earlier will have a destination MAC rewrite tlv on the filter switch, but no such tlv appears on the core or delivery switch.

As an alternative, drop into the bash of the filter switch to check the flow and destination MAC rewrite.

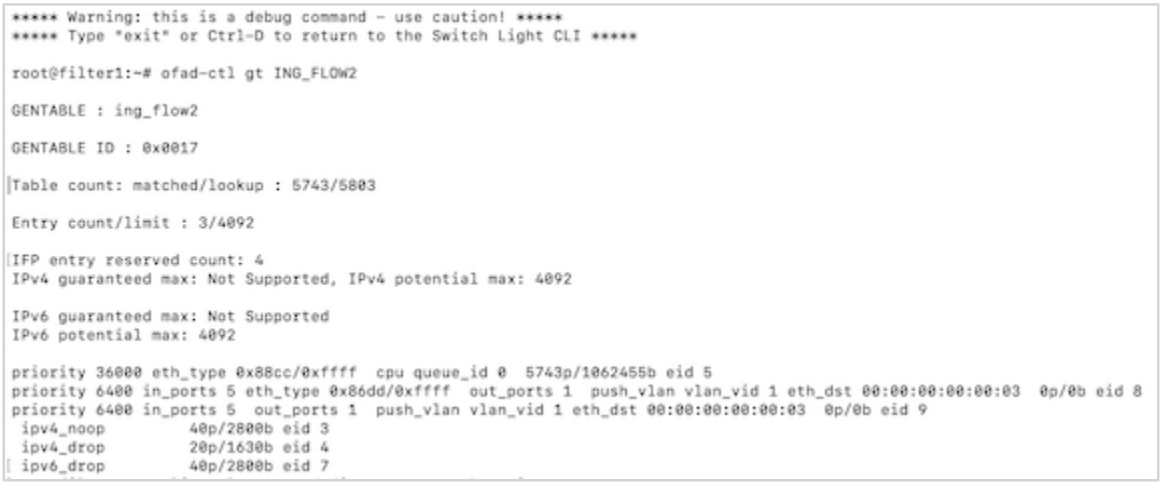
Use the following commands for the ZTN CLI of the filter switch.

```
(config)# connect switch filter1
(ztn-config) debug admin
filter1> enable
filter1# debug bash
```

The following command prints the flow table of the filter switch.

```
root@filter1:~# ofad-ctl gt ING_FLOW2
```

Figure 2-100: Filter Switch Flow Table



The following command shows the policy flow from the filter switch to the delivery switch. The filter switch will have the assigned destination MAC in the match-field.

```
(config)# show policy-flow
# Policy Name Switch                                Pkts Bytes Pri  T Match
 Instructions
-|-----------|-----------------------------------|----|-----|----|-|----------------------
--|------------------|
1 p1          core1 (00:00:52:54:00:15:94:88)     0    0     6400 1 eth-type ipv6,vlan-vid 1
 apply: name=p1 output: max-length=65535, port=2
2 p1          core1 (00:00:52:54:00:15:94:88)     0    0     6400 1 vlan-vid 1
 apply: name=p1 output: max-length=65535, port=2
3 p1          delivery1 (00:00:52:54:00:00:11:d2) 0    0     6400 1 vlan-vid 1
 apply: name=p1 output: max-length=65535, port=6
4 p1          delivery1 (00:00:52:54:00:00:11:d2) 0    0     6400 1 eth-type ipv6,vlan-vid 1
 apply: name=p1 output: max-length=65535, port=6
5 p1          filter1 (00:00:52:54:00:d5:2c:05)   0    0     6400 1
   apply: name=p1 push-vlan: ethertype=802.1Q (33024),set-field: match-field/type=vlan-vid,
 match-field/vlan-tag=1,output: max-length=65535, port=1,set-field: match-field/eth-address=00
:00:00:00:00:03 (XEROX), match-field/type=eth-dst
6 p1          filter1 (00:00:52:54:00:d5:2c:05)   0    0     6400 1 eth-type ipv6
   apply: name=p1 push-vlan: ethertype=802.1Q (33024),set-field: match-field/type=vlan-vid,
 match-field/vlan-tag=1,output: max-length=65535, port=1,set-field: match-field/eth-address=00
:00:00:00:00:03 (XEROX), match-field/type=eth-dst
```

Considerations

1. The destination MAC rewrite cannot be used on the filter interface where timestamping is enabled.
2. The destination MAC rewrite will not work when the filter interface is configured as a receive-only tunnel interface.

2.3.10.4 Forward Slashes in Interface Names

DANZ Monitoring Fabric (DMF) supports using forward slashes (/) in interface names to aid in managing interfaces in the DMF fabric. For example, when:

- Defining the SPAN device name and port numbers which generally contain a forward slash (eth2/1/1) in the name for easy port identification.
- Using separate SPAN sessions for Tx and Rx traffic, when there are multiple links from a device to a filter switch.

The following is the comprehensive list of DMF interfaces supporting the use of a forward slash:

filter interface	unmanaged service interface
filter interface group	recorder node interface
delivery interface	MLAG interface
delivery interface group	LAG interface
filter-and-delivery interface	GRE tunnel interface
PTP interface	VXLAN tunnel interface
managed service interface	



Note: An interface name cannot start with a forward slash. However, multiple forward slashes are allowed while adhering to the maximum allowed length limitation.

Configuration

The configuration of DMF filter interfaces remains unchanged. This feature relaxes the existing naming convention by allowing a forward slash to be a part of the name.

The following are several examples:

For switch interfaces, for any of the roles: **`both-filter-and-delivery`**, **`delivery`**, **`filter`**, **`ptp`**, **`service`**

```
dmf-controller-1(config-switch-if)# role role interface-name a/b/c
```

For filter and delivery interface groups:

```
dmf-controller-1(config)# filter-interface-group a/b/c
dmf-controller-1(config)# delivery-interface-group a/b/c
```

Adding interfaces or interface groups to a policy:

```
dmf-controller-1(config-policy)# filter-interface f1/a/b
dmf-controller-1(config-policy)# filter-interface-group f/a/b
dmf-controller-1(config-policy)# delivery-interface d1/a/b
dmf-controller-1(config-policy)# delivery-interface-group d/a/b
```

Recorder Node interface:

```
dmf-controller-1(config)# recorder-fabric interface a/b/c
```

For a managed service:

```
dmf-controller-1(config-managed-srv-flow-diff)# l3-delivery-interface a/b/c
```

MLAG interface:

```
dmf-controller-1(config-mlag-domain)# mlag-interface a/b/c
dmf-controller-1(config-mlag-domain-if)# role delivery interface-name a/b/c
```

LAG interface:

```
dmf-controller-1(config-switch)# lag-interface a/b/c
```

GRE tunnel interface:

```
dmf-controller-1(config-switch)# gre-tunnel-interface a/b/c
```

VXLAN tunnel interface:

```
dmf-controller-1(config-switch)# vxlan-tunnel-interface a/b/c
```

Show Commands

There are no new **show** commands. The existing **show running-config** and **show this** commands for the configurations mentioned earlier should display the interface names without any issue.

2.3.10.5 Configure sFlow on a Filter and Delivery Interface

A DANZ Monitoring Fabric (DMF) interface used by a DMF policy as both a filter and a delivery interface is known as a filter-and-delivery interface. Filter-and-delivery interfaces support configuring sFlow®* in the DMF Controller. DMF enables sFlow by default on filter-and-delivery interfaces.

Perform the following steps using the CLI:

1. To disable sFlow on a filter-and-delivery interface, use the DMF Controller CLI and enter the configure submode.

```
dmf-controller> enable
dmf-controller# configure
dmf-controller(config)#
```

2. Enter the submode of the switch, followed by the submode of the switch interface where the filter-and-delivery interface is already defined or is to be defined.

```
dmf-controller(config)# switch sw1
dmf-controller(config-switch)# interface Ethernet1/1
dmf-controller(config-switch-if)#
```

3. Use the **role** command to define or edit the filter-and-delivery interface. Specify the optional **no-sflow** token after the interface name to disable sFlow on the filter-and-delivery interface.

```
dmf-controller(config-switch-if)# role both-filter-and-delivery interface-name fd1
 no-analytics             Disable interface analytics (optional)
 no-sflow                 Disable interface sFlow (optional)
 no-strip-vxlan           Enable/disable VxLAN header stripping (optional)
 no-vlan-preservation     Disable VLAN preservation config for this delivery interface
 (optional)
 rewrite                  Select header value(s) to rewrite
 strip-no-vlan            Customize Strip VLAN tag on this delivery interface (optional)
 strip-one-vlan           Customize Strip VLAN tag on this delivery interface (optional)
 strip-second-vlan        Customize Strip VLAN tag on this delivery interface (optional)
 strip-two-vlan           Customize Strip VLAN tag on this delivery interface (optional)
 strip-vxlan              Enable/disable VxLAN header stripping (optional)
<Command-end>             <cr>: Associate filter-and-delivery role with switch interface
 ;                        command separator
 |                        pipe to command
 >                        redirect

dmf-controller(config-switch-if)# role both-filter-and-delivery interface-name fd1 no-sflow
```

If sFlow is disabled, apply the **role** command without the optional **no-flow** token to enable it and restore the default sFlow behavior on the filter-and-delivery interface.



Important: Be certain to include and preserve any other optional tokens since the **role** command replaces the role config for the interface.

```
dmf-controller(config-switch-if)# role both-filter-and-delivery interface-name fd1
```

* sFlow® is a registered trademark of Inmon Corp.

2.3.10.6 Configure Interface Groups

Create an interface group consisting of one or more filter or delivery interfaces. It is often easier to refer to an interface group when creating a policy than to identify every interface to which the policy applies explicitly.

Use an address group in multiple policies, referring to the IP address group by name in match rules. If no subnet mask is provided in the address group, it is assumed to be an exact match. For example, in an IPv4 address group, the absence of a mask implies a mask of /32. For an IPv6 address group, the absence of a mask implies a mask of /128.

Identify only a single IP address group for a specific policy match rule. Address lists with both **src-ip** and **dst-ip** options cannot exist in the same match rule.

The following example illustrates the configuration of two interface groups: a filter interface group ***TAP-PORT-GRP***, and a delivery interface group ***TOOL-PORT-GRP***.

```
controller-1(config-switch)# filter-interface-group TAP-PORT-GRP
controller-1(config-filter-interface-group)# filter-interface TAP-PORT-1
controller-1(config-filter-interface-group)# filter-interface TAP-PORT-2
controller-1(config-switch)# delivery-interface-group TOOL-PORT-GRP
controller-1(config-delivery-interface-group)# delivery-interface TOOL-PORT-1
controller-1(config-delivery-interface-group)# delivery-interface TOOL-PORT-2
```

To view information about the interface groups in the DMF fabric, enter the **`show filter-interface-group`** command, as in the following examples:

Filter Interface Groups

```
controller-1(config-filter-interface-group)# show filter-interface-group
! show filter-interface-group TAP-PORT-GRP
# Name         Big Tap IF Name            Switch IF Name    Direction  Speed     State   VLAN  Tag
-|------------|--------------------------|-----------------|----------|---------|-------|-----|--------|
1 TAP-PORT-GRP TAP-PORT-1 DMF-CORE-SWITCH-1 ethernet17      rx         100Gbps   up      0
2 TAP-PORT-GRP TAP-PORT-2 DMF-CORE-SWITCH-1 ethernet18      rx         100Gbps   up      0
controller1(config-filter-interface-group)#
```

Delivery Interface Groups

```
controller1(config-filter-interface-group)# show delivery-interface-group
! show delivery-interface-group DELIVERY-PORT-GRP
# Name              Big Tap IF Name Switch IF Name        Direction  Speed     Rate   limit     State Strip Forwarding Vlan
-|-----------------|---------------|---------------------|----------|---------|------|---------|-----|---------------------|
1 TOOL-PORT-GRP TOOL-PORT-1 DMF-DELIVERY-SWITCH-1 ethernet15 tx      10Gbps                      up    True
2 TOOL-PORT-GRP TOOL-PORT-2 DMF-DELIVERY-SWITCH-1 ethernet16 tx      10Gbps                      up    True
controller-1(config-filter-interface-group)#
```

2.3.10.7 Configure Error Disable State

Error Disable State for Port Flapping prevents policy churn by automatically placing switch interfaces with frequent flapping into an error-disabled state, effectively performing an automatic administrative shutdown. The feature allows for automatically recovering these interfaces after a specified time. It reduces the risk of lost packets caused by continuous re-computation of DANZ Monitoring Fabric (DMF) policies due to flapping interfaces. By default, Error Disable State for Port Flapping is disabled.

Global Configuration

This feature is configurable at the global level.

To configure it, run the **errdisable** command from the **config** submode.

```
dmf-controller(config)# errdisable
dmf-controller(config-errdisable)#
detect-duration    enable-feature     flap-count         recovery-interval
```

By default, this feature is disabled; use the **enable-feature** command to enable it under the **config-errdisable** submode.

```
dmf-controller(config-errdisable)# enable-feature
```

Conversely, to deactivate it, use the **no** option of **enable-feature**.

```
dmf-controller(config-errdisable)# no enable-feature
```

Flap-count: Specifies the number of flaps required within the detect-duration period to set the interface into the error-disabled state. The default is **5** flaps.

```
dmf-controller(config-errdisable)# flap-count
Flap count     Configure flap count. Range 1..1000 (default 5)
dmf-controller(config-errdisable)# flap-count 20
```

Detect-duration: Specifies the duration during which the required flap count must be reached to set the interface into an error-disabled state. The default is **10** seconds.

```
dmf-controller(config-errdisable)# detect-duration
Detect duration (secs)     Configure detect duration in seconds. Range 1..1800 (default 10s)
dmf-controller(config-errdisable)# detect-duration 30
```

Recovery-interval: The **recovery-interval** indicates how much time will pass before bringing up the interface when the automatic administrative shutdown action shuts it down. The default is **3600** seconds.

```
dmf-controller(config-errdisable)# recovery-interval
Recovery interval (secs)     Configure the recovery interval in seconds. Range 30..86400 (default
 3600s)
dmf-controller(config-errdisable)# recovery-interval 1200
```

Use the **show this** command to review the settings:

```
dmf-controller(config-errdisable)# show this
! errdisable
errdisable
  detect-duration 30
  enable-feature
  flap-count 20
  recovery-interval 1200
```

Local Configuration

This feature allows disabling the Error Disable feature on a per-interface basis. This prevents a specific interface from being automatically error-disabled due to port flapping.

To disable Error Disable for an interface:

Enter the interface configuration mode **config-switch-if** using the following commands:

```
dmf-controller(config)# switch switch-name
dmf-controller(config-switch)# interface interface-name
dmf-controller(config-switch-if)#
```

Use the **errdisable** command to access the interface's Error Disable settings which enters into **errdisable** submode.

Execute the **disable** command.

Disable the feature on the interface level using the **disable** option under **config-switch-if-errdisable**.

```
dmf-controller(config-switch-if)# errdisable
dmf-controller(config-switch-if-errdisable)# disable
```

Quick Interface Recovery

When an interface is in the error-disabled state, use the **disable** command to immediately bring the interface back up instead of waiting for the recovery action to be initiated. This action overrides the configured recovery interval and restores the interface's functionality.

2.3.10.8 Show Commands - Error State

When an interface is in error-disabled state, view the details using the **show fabric warnings errdisable-warning** command. The output lists the switch and the error-disabled interface name with a detailed message.

```
dmf-controller# show fabric warnings errdisable-warning
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Error disable warnings  ~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~
# Switch IF Name    Warning message

-|------|----------|--------------------------------------------------------------------------
--------|
1 core1  ethernet12 Interface ethernet12 on switch core1 has been error-disabled due to port flaps
```

Use the **show errdisable** command to obtain the following details:

- Global settings for the error-disable port flapping feature, including the maximum flap count, monitor duration, recovery interval, and status.
- A list of currently error-disabled interfaces and their switch and interface names.
- A list of interfaces where the error-disable feature has been specifically disabled.

```
dmf-controller# show errdisable
Max flap count                    : 20
Duration to monitor max flaps (sec) : 30
Recovery interval (sec)           : 1200
Status                            : enabled
~ Interfaces in Error-Disable State ~
# Switch IF Name
-|------|----------|
1 core1  ethernet12
~ Excluded Interfaces from Error-Disable ~
# Switch IF Name
-|------|----------|
1 core1  ethernet14
```

Use the **show switch *switch-name* interface *interface-name* errdisable** command to view more details related to the error-disable interface. The output lists the switch name, interface name, state, the reason for the port shutdown, shutdown time, time remaining until its recovery, and expected recovery time.

```
dmf-controller# show switch core1 interface ethernet12 errdisable
# Switch IF Name    State         Shutdown Reason Shutdown Time           Time Until Recovery(sec)
 Expected Recovery Time
-|------|----------|-------------|---------------|----------------------|----------------
--------|----------------------|
1 core1  ethernet12 port-disabled port-flaps      2025-02-26 20:46:29 UTC 973
 2025-02-26 21:46:29 UTC
```

2.3.10.9 Troubleshooting

For troubleshooting scenarios where **errdisable** is configured but an interface has not been disabled despite flapping, the **show flaps port *switch-name interface-name*** command is useful. This command provides the following information:

- **Total Flap Count:** The cumulative number of detected flap events for the specified interface.
- **Recent Flap History:** The last 10 flap events detail the precise start and end timestamps of the interface's down and up transitions.



Note: To identify true interface flapping, DMF establishes a baseline by observing a brief period of initial up/down transitions. The **show flaps port *switch-name interface-name*** command registers and displays actual flapping events only after establishing this baseline.

```
# show flaps port core1 ethernet12
~~~~~~~~ Selected Port  ~~~~~~~~
Port Name           : ethernet12
Switch Name         : core1
Total Flaps on Port : 3
~~~~~~~~~~~~~~~~ Port Flap Events ~~~~~~~~~~~~~~~~
# Start Time of Flap      End Time of Flap
-|----------------------|----------------------|
1 2025-02-27 19:00:33 UTC 2025-02-27 19:00:34 UTC
2 2025-02-27 19:00:30 UTC 2025-02-27 19:00:31 UTC
3 2025-02-27 19:00:28 UTC 2025-02-27 19:00:29 UTC
```

2.3.10.10 Overriding the Default Configuration for a Switch

By default, each switch inherits its configuration from the DANZ Monitoring Fabric (DMF) Controller. Use the following pages of the **Configure Switch** dialog to override the following configuration options for a specific switch.

- Info
- Clock
- SNMP
- SNMP traps
- Logging
- TACACS
- sFlow®*
- LAG enhanced hash

* sFlow® is a registered trademark of Inmon Corp.

2.3.10.10.1 CLI Configuration

To use the CLI to manage switch configuration, enter the following commands to enter the **config-switch** submode.

```
controller-1(config)# switch switch-name
```

Replace the ***switch-name*** with the alias previously assigned to each switch during installation, as in the following example.

```
controller-1(config)# switch DMF-SWITCH-1
controller-1(config-switch)#
```

From this submode, configure the specific switch and override the default configuration pushed from the DANZ Monitoring Fabric (DMF) Controller to the switch.

The ***DANZ Monitoring Fabric Deployment Guide*** provides detailed instructions on overriding the switch's default configuration.

2.4 Switch Light CLI Operational Commands

As a result of upgrading the Debian distribution to Bookworm, the original Python CLI (based on python2) was removed, as the interaction with the DANZ Monitoring Fabric (DMF) is performed mainly from the Controller.

However, several user operations involve some of the commands used on the switch. These commands are implemented in the new CLI (based on python3) in Switch Light in the Bookworm Debian distribution.

The Zero-Trust Network (ZTN) Security CLI is the default shell when logged into the switch.



Note: The following commands are only available on Arista Switch platforms running Switch Light OS.

Operational Commands

After connecting to the switch and from the DMF Controller, use the **`debug admin`** command to enter the switch admin CLI from the ZTN CLI.

Enter the **`exit`** command to leave the switch admin CLI, as illustrated in the following example.

```
DMF-CONTROLLER# connect switch dmf-sw-7050sx3-1
Switch Light OS SWL-OS-DMF-8.6.x(0), 2024-05-16.08:26-17f56f6
Linux dmf-sw-7050sx3-1 4.19.296-OpenNetworkLinux #1 SMP Thu May 16 08:35:25 UTC 2024 x86_64
Last login: Tue May 21 10:39:05 2024 from 10.240.141.151

Switch Light ZTN Manual Configuration. Type help or ? to list commands.

(ztn-config) debug admin
(admin)
(admin) exit

(ztn-config)
```

Help

The following commands are available under the admin shell.

```
(admin) help

Documented commands (type help <topic>):
========================================
EOF  copy  exit  help  ping  ping6  quit  reboot  reload  show
```

Ping

Use the **`ping`** command to test a host's accessibility using its IPV4 address.

```
(admin) ping 10.240.141.151
PING 10.240.141.151 (10.240.141.151) 56(84) bytes of data.
64 bytes from 10.240.141.151: icmp_seq=1 ttl=64 time=0.238 ms
64 bytes from 10.240.141.151: icmp_seq=2 ttl=64 time=0.206 ms
64 bytes from 10.240.141.151: icmp_seq=3 ttl=64 time=0.221 ms
64 bytes from 10.240.141.151: icmp_seq=4 ttl=64 time=0.161 ms

--- 10.240.141.151 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3049ms
rtt min/avg/max/mdev = 0.161/0.206/0.238/0.028 ms
```

Ping6

Use the **ping6** command to test a host's accessibility using its IPV6 address.

```
(admin) ping6 fe80::3673:5aff:fefb:9dec
PING fe80::3673:5aff:fefb:9dec(fe80::3673:5aff:fefb:9dec) 56 data bytes
64 bytes from fe80::3673:5aff:fefb:9dec%ma1: icmp_seq=1 ttl=64 time=0.490 ms
64 bytes from fe80::3673:5aff:fefb:9dec%ma1: icmp_seq=2 ttl=64 time=0.232 ms
64 bytes from fe80::3673:5aff:fefb:9dec%ma1: icmp_seq=3 ttl=64 time=0.218 ms
64 bytes from fe80::3673:5aff:fefb:9dec%ma1: icmp_seq=4 ttl=64 time=0.238 ms

--- fe80::3673:5aff:fefb:9dec ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3069ms
rtt min/avg/max/mdev = 0.218/0.294/0.490/0.113 ms
```

Copy Tech-Support Data

Use the **copy tech-support data** command to collect the switch support bundle. The tech support command executed from the Controller collects support bundles from all the switches in the fabric.

To collect tech support data for an individual switch, only run the **copy tech-support data** command from that switch. Perform this action when the switch isn't accessible from the Controller or when collecting a support bundle only for that switch.

```
(admin) copy tech-support data
Writing /mnt/onl/data/tech-support_240521104337.txt.gz...
tech-support_240521104337.txt.gz created in /mnt/onl/data/
(admin)
```

Show Commands

The following **show** commands are available under the admin shell.

Show Clock

```
(admin) show clock
Tue May 21 10:46:20 2024
(admin)
```

Show NTP

```
(admin) show ntp
remote                                  refid           st t when poll  reach delay    offset
 jitter
==============================================================================================
======
 10.130.234.12                          .STEP.          16 u    - 1024    0   0.0000   0.0000
 0.0002
*time2.google.com                       .GOOG.           1 u  649 1024  377  40.0469   1.6957
 0.4640
+40.119.6.228                           25.66.230.1      3 u  601 1024  377  50.6083  -0.5323
 7.8069
(admin)
```

Show Controller

The **show controller** command displays all the configured Controllers and their connection status and role.

```
(admin) show controller
IP:Port              Proto State          Role     #Aux
10.240.141.151:6653  tcp   CONNECTED      ACTIVE      2
10.240.189.233:6653  tcp   CONNECTED      STANDBY     2
     127.0.0.1:6653  tcp   CONNECTED      STANDBY     1
(admin)
```

Use the **history** and **statistics** options in the **show controller** command to obtain additional information.

```
(admin) show controller history | statistics
(admin)
```

The **show controller history** command displays the history of controller-to-switch connections and disconnections.

```
(admin) show controller history
Mon May 20 15:53:42 2024 tcp:127.0.0.1:6653:0 - Connected
Mon May 20 15:53:43 2024 tcp:127.0.0.1:6653:1 - Connected
Mon May 20 15:54:46 2024 tcp:127.0.0.1:6653:1 - Disconnected
Mon May 20 15:54:46 2024 tcp:127.0.0.1:6653:0 - Disconnected
Mon May 20 08:57:07 2024 tcp:127.0.0.1:6653:0 - Connected
Mon May 20 08:57:07 2024 tcp:127.0.0.1:6653:1 - Connected
Mon May 20 08:57:07 2024 tcp:10.240.141.151:6653:0 - Connected
Mon May 20 08:57:07 2024 tcp:10.240.141.151:6653:1 - Connected
Mon May 20 08:57:07 2024 tcp:10.240.141.151:6653:2 - Connected
Mon May 20 11:16:07 2024 tcp:10.240.189.233:6653:0 - Connected
Mon May 20 11:16:19 2024 tcp:10.240.189.233:6653:1 - Connected
Mon May 20 11:16:19 2024 tcp:10.240.189.233:6653:2 - Connected
(admin)
```

The **`show controller statistics`** command displays connection statistics, including keep-alive timeout, timeout threshold count, and other important information, as shown in the following example.

```
(admin) show controller statistics
Connection statistics report
Outstanding async op count from previous connections: 0
Stats for connection tcp:10.240.141.151:6653:0:
    Id: 131072
    Auxiliary Id: 0
    Controller Id: 0
    State: Connected
    Keepalive timeout: 2000 ms
    Threshold: 3
    Outstanding Echo Count: 0
    Tx Echo Count: 46438
    Messages in, current connection: 52887
    Cumulative messages in: 52887
    Messages out, current connection: 52961
    Cumulative messages out: 52961
    Dropped outgoing messages: 0
    Outstanding Async Operations: 0
Stats for connection tcp:10.240.189.233:6653:0:
    Id: 112066561
    Auxiliary Id: 0
    Controller Id: 1
    State: Connected
    Keepalive timeout: 2000 ms
    Threshold: 3
    Outstanding Echo Count: 0
    Tx Echo Count: 42269
    Messages in, current connection: 43108
    Cumulative messages in: 43108
    Messages out, current connection: 43114
    Cumulative messages out: 43114
    Dropped outgoing messages: 0
    Outstanding Async Operations: 0
(admin)
```

The **show log** command displays log messages from the Syslog file.

```
(admin) show log
2024-05-20T15:53:04+00:00 localhost syslog-ng[3787]: NOTICE syslog-ng starting up; version='3.38
.1'
2024-05-20T15:52:51+00:00 localhost kernel: NOTICE Linux version 4.19.296-OpenNetworkLinux
 (bsn@sbs3) (gcc version 12.2.0 (Debian 12.2.0-14)) #1 SMP Thu May 16 08:35:25 UTC 2024
2024-05-20T15:52:51+00:00 localhost kernel: INFO Command line:  reboot=p acpi=on Aboot=Aboot-
norcal6-6.1.10-14653765 platform=magpie sid=Calpella console=ttyS0,9600n8 tsc=reliable
 pcie_ports=native pti=off reassign_prefmem amd_iommu=off onl_mnt=/dev/mmcblk0p1 quiet=1
 onl_platform=x86-64-arista-7050sx3-48yc12-r0 onl_sku=DCS-7050SX3-48YC12
2024-05-20T15:52:51+00:00 localhost kernel: INFO BIOS-provided physical RAM map:
….
….
2024-05-21T10:40:31-07:00 dmf-sw-7050sx3-1 systemd[1]: INFO Starting logrotate.service - Rotate
 log files...
2024-05-21T10:40:32-07:00 dmf-sw-7050sx3-1 systemd[1]: INFO logrotate.service: Deactivated
 successfully.
2024-05-21T10:40:32-07:00 dmf-sw-7050sx3-1 systemd[1]: INFO Finished logrotate.service - Rotate
 log files.
2024-05-21T10:45:32-07:00 dmf-sw-7050sx3-1 systemd[1]: INFO Starting logrotate.service - Rotate
 log files...
2024-05-21T10:45:33-07:00 dmf-sw-7050sx3-1 systemd[1]: INFO logrotate.service: Deactivated
 successfully.
2024-05-21T10:45:33-07:00 dmf-sw-7050sx3-1 systemd[1]: INFO Finished logrotate.service - Rotate
 log files.
(admin)
```

Reboot and Reload

Use the **reboot** and **reload** commands to either reboot or reload the switch, as needed.

```
(admin) help reboot

Reboot the switch.

(admin) help reload

Reload the switch.

(admin) reboot
Proceed with reboot [confirm]?

(admin) reload
Proceed with reload [confirm]?
(admin)
```

Chapter 3

Managing DMF Policies

This chapter describes the policies to work and configure in the DANZ Monitoring Fabric (DMF).

3.1 Overview

A policy selects the traffic to be copied from a production network to one or more tools for analysis. To define a policy, identify the traffic source(s) (filter interfaces), the match rules to select the type of traffic, and the destination tool(s) (delivery interfaces). The DANZ Monitoring Fabric (DMF) Controller automatically forwards the selected traffic based on the fabric topology. Define match rules to select interesting traffic for forwarding to the tools connected to the specified delivery interfaces. Users can also send traffic to be processed by a managed service, such as time stamping, slicing, or deduplication, on a DMF service node. Forward the output from the service node to the appropriate tool for analysis.

While policies can be simple, they can also be more complicated when optimizing hardware resources, such as switching TCAM space. Also, DMF provides different switching modes to optimize policies based on use cases and switch capabilities. Arista Networks recommends planning the switching mode before configuring policies in a production deployment.

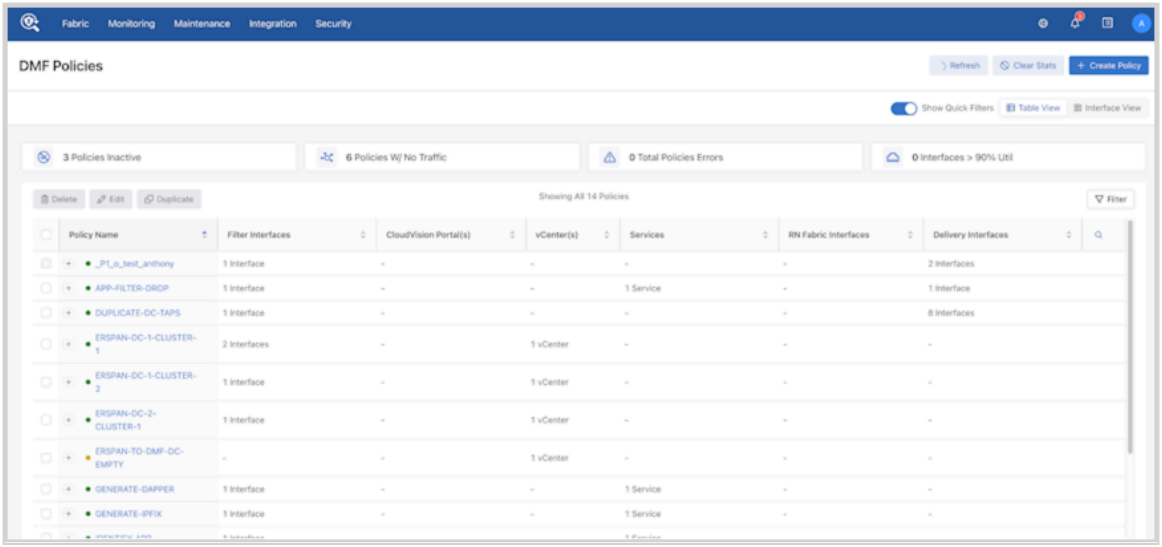
For further information, refer to the chapter Advanced Policy Configuration.

3.2 DMF Policies Page

Overview

While retaining all information from the previous version, the new policy page features a new layout and design and offers additional functionality for easier viewing, monitoring, and troubleshooting of policies.

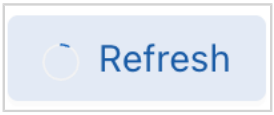
Figure 3-1: DMF Policies



Header Action Items

- Refresh Button

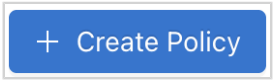
Figure 3-2: Refresh Button



The page refreshes every 60 seconds automatically. Select **Refresh** to manually refresh the page.

- Create Policy Button

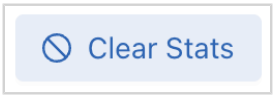
Figure 3-3: Create Policy Button



Select **+ Create Policy** to open the policy creation page.

- Clear Stats Button

Figure 3-4: Clear Stats Button

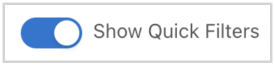


Select **Clear Stats** to clear all DMF interface's runtime stats.

Quick Filters

- Show Quick Filters Button

Figure 3-5: Show Quick Filters



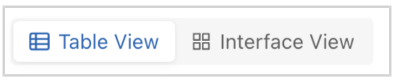
By default, the feature is toggled **on** and displays four quick filter options. When toggled **off**, the four quick filters are no longer displayed.

Figure 3-6: Four Filter Options

Four quick filter cards display the policy counts that meet the filter criteria and the filter name. The quick filter cards support multi-select.

- Radio Buttons

Figure 3-7: Table View / Interface View



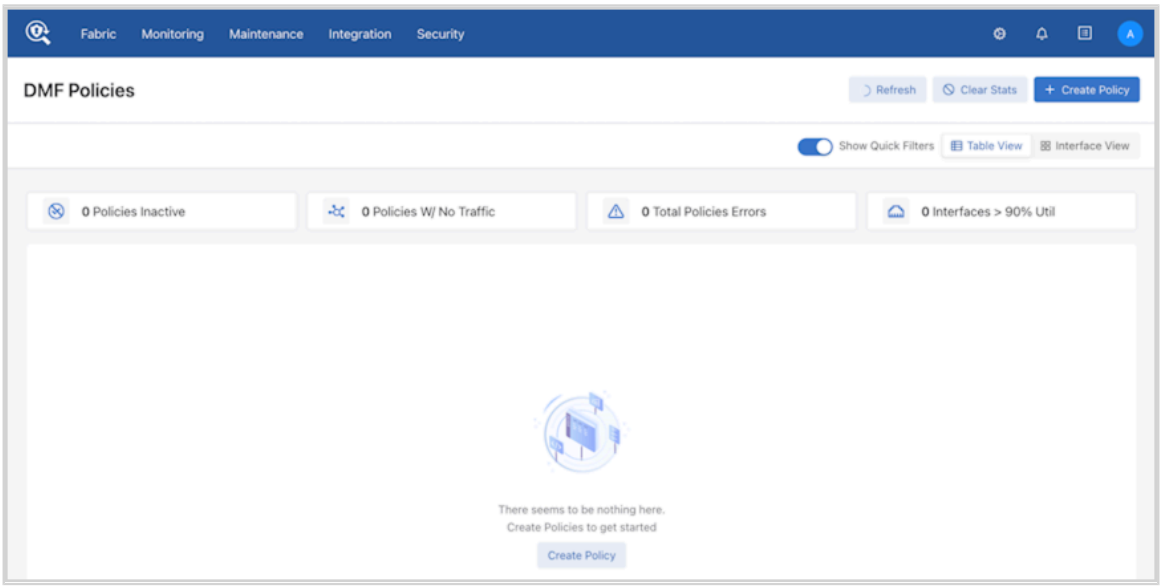
Switch page views between **Table View** and **Interface View**. Refer to the **Table View** and **Interface View** sections below for more information.

Table View

The table view is the default landing view of the **Policies** Page.

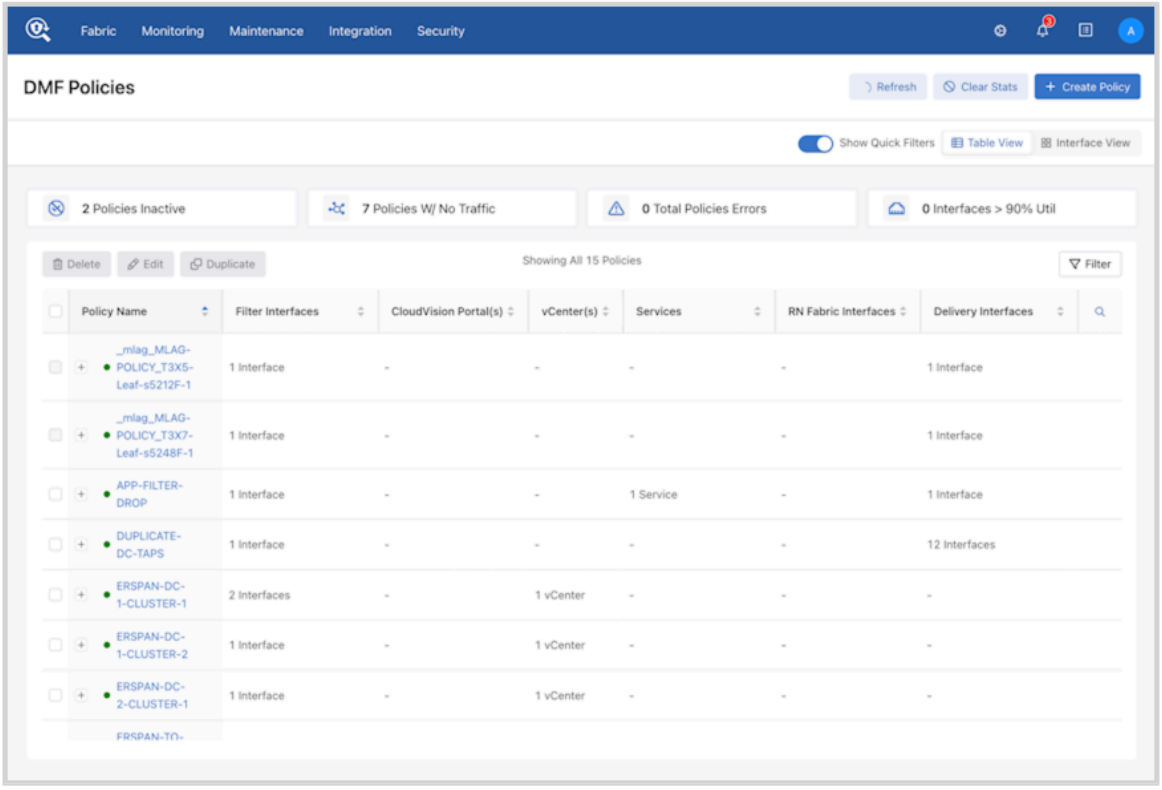
The page displays an empty table with the **Create Policy** control when no configured policies exist.

Figure 3-8: DMF Policies



Conversely, when configured policies exist, the table view displays the list of policies.

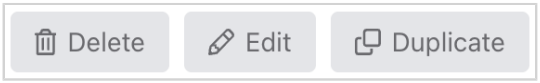
Figure 3-9: List of Policies



Action Buttons

Several buttons in the policy table provide quick access to corresponding functionality. These are:

Figure 3-10: Action Buttons



Delete

- Disabled by default (when no policies are selected).
- Enabled when one or more policies are selected.
- Used to delete selected policies.

Edit

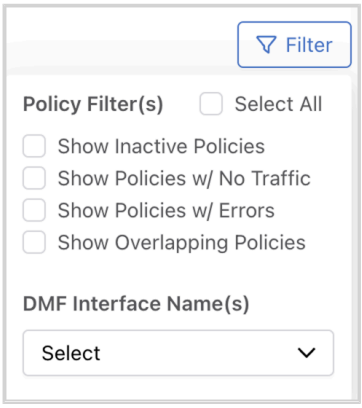
- Disabled by default (when no policies are selected).
- Enabled only when a policy is selected.
- Navigate to the editing workflow (the new policy edit workflow).

Duplicate

- Disabled by default (when no policy is elected).
- Enabled only when one policy is selected.
- Navigate to the create policy workflow (the new policy create workflow) with an empty name input field while retaining the same settings from the selected policy.

Table View Filters

Figure 3-11: Filter Views



Select **Filter** to open the filter menu.

Policy Filter(s)

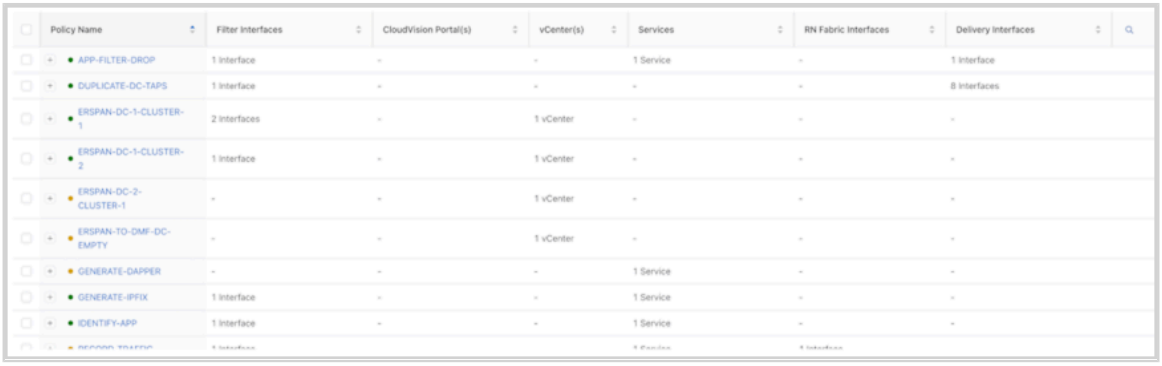
- There are four quick policy filters. The first three filters overlap with the quick filters; thus, enabling or disabling them will trigger changes to the quick filter control.

DMF Interface Name(s)

- Filters out policies by DMF interfaces that are selected from the drop-down list.
- Searchable
- Allows multiple selections applying OR logic.

Policies Table

Figure 3-12: Policy Table



The **Policy** table displays all policies; each column shows the number of interfaces and services corresponding to that policy.

Figure 3-13: Search



Table Search

The **Policy** table supports search functionality. Select the **magnifying glass** icon in the last column to activate the search input fields and search the results by the context of each column.

Table Search: The Policy table supports search functionality. Select the **magnifying glass** icon in the last column to activate the search input fields. Search results by the context of each column.

Figure 3-14: Table Search

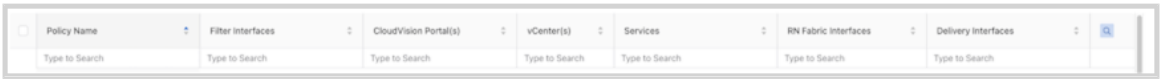


Figure 3-15: Expand Icon



Expand Policy + Icon

Hidden for an unconfigured policy. Select the expand **+** icon to view the policy's interfaces and services information.

Figure 3-16: Expanded View Example

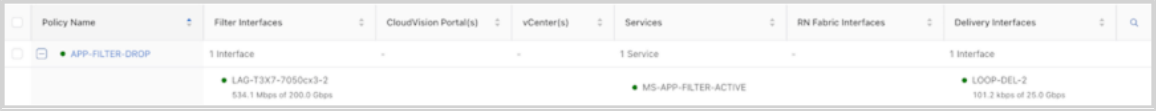
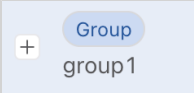


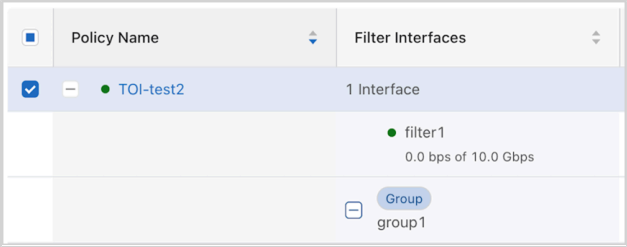
Figure 3-17: Expand Group



Interfaces Group Expand + Icon

For policies configured with an interface group, an expand **+** icon with group displays by default. Select the group expand **+** icon to view the detailed information on the interfaces belonging to that group.

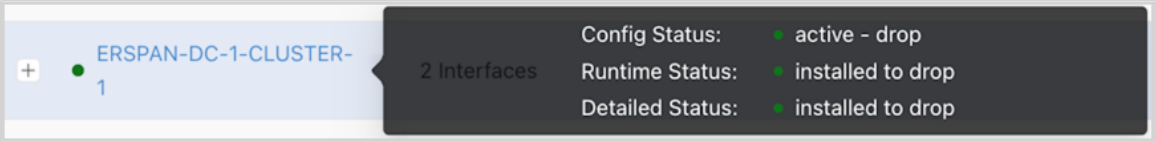
Figure 3-18: Filter Interface Details



Policy Name Tooltip

Hovering over policy names displays the tooltips for the policy, including Configuration / Runtime / Details state.

Figure 3-19: Tooltip



Policy Error Icon

Figure 3-20: Policy Error Icon



Policies with errors will display this icon after the policy name.

Figure 3-21: Error with Policy Name

Selecting the **error** icon will display an error window with detailed information.

Figure 3-22: Detailed Error Information



Checkbox

Figure 3-23: Checkbox

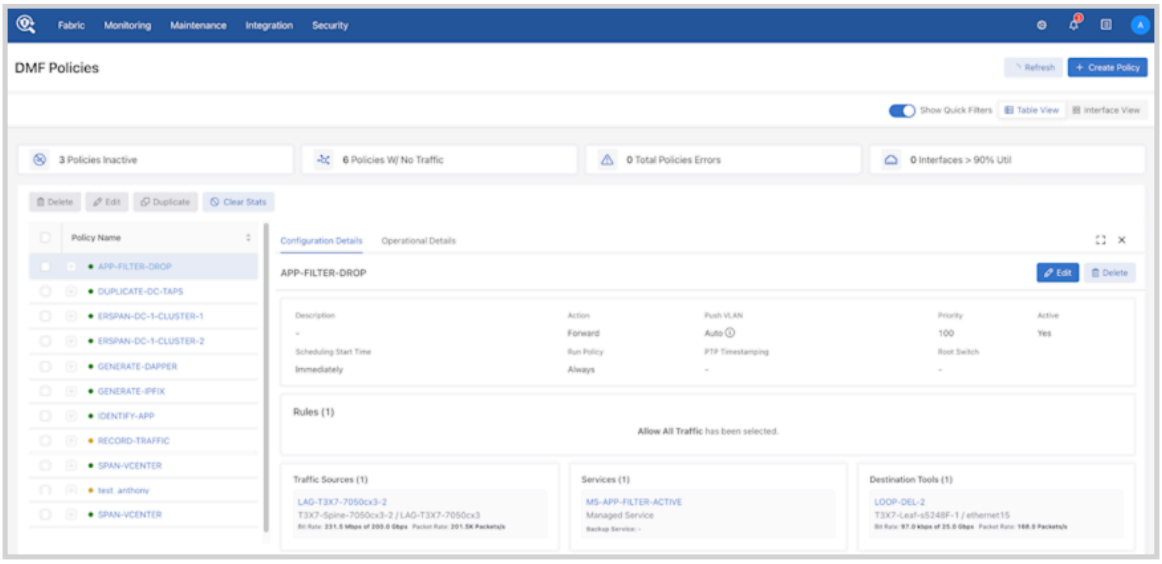
Disabled for unconfigured policies. Use the **checkbox** to select a policy and the applicable function buttons (described above) as required.

Table Interaction

- All columns support sorting.
- Selecting a **policy name** opens the policy table split view. The table on the left displays the policy names. The table on the right provides two tabs showing **Configuration** and **Operational Details**.
- Select a policy name from the **Policy Name** list to view its configuration or operational details in the split view.

- Use the ⛶ icon to view the information in full-screen mode or the **X** icon to close the split view and return to the table view.

Figure 3-24: DMF Policies

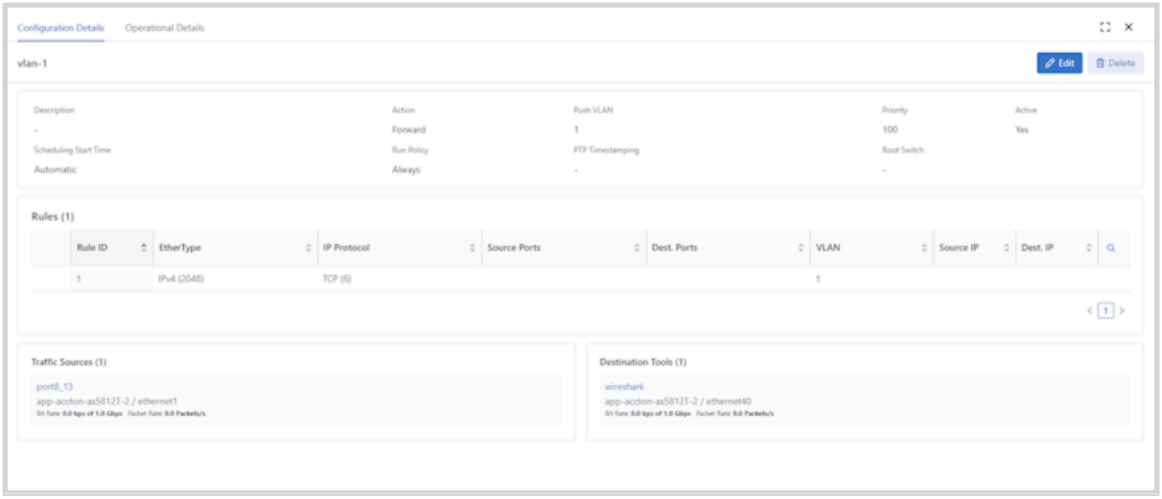


Configuration Details

Access the **Configuration Details** tab by selecting a policy in either **Table View** or **Interface View**. This tab displays all of the configured settings for the selected policy.

The top row of the **Configuration Details** tab displays the selected policy name and an **Edit** and **Delete** control. **Edit** opens the **Edit Policy** configuration page with policy information prefilled, and **Delete** opens a confirmation dialog window before deleting a policy. The default **Table View** opens after deleting a policy.

Figure 3-25: Configuration Details



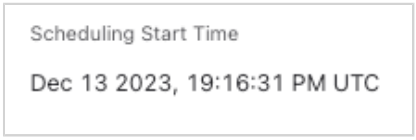
The second component of the Configuration Details is the Quick Facts box. This component displays the Description, Action, Push VLAN, Priority, Active, Scheduling Start Time, Policy Run Duration, PTP Timestamping, and Root Switch values.

- **Description**: An info icon shows the entire description in a tooltip.
- **Action**: Forward, Drop, Capture, or None.
- **Active**: Policy active status, Yes or No.

  Scheduling Start Time: Either **Automatically** or the DateTime it is scheduled to start, in terms of the current Time Zone configured on the DMF. When setting DateTime to **Now** during policy creation, the time of creation will be the Scheduling start time.

  - **Automatic**: The policy will always run. There's no expiration.
  - **Now**: The policy starts from now, and duration and packet expiration may apply. The policy runs from now with no expiration.

    Figure 3-26: Start Time

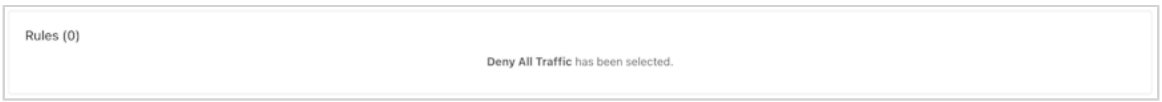
- **Run Policy**: The duration the policy should run. The default value is **Always**. Set a time limit (i.e., 4 hours) or a packet limit (i.e., 1,000 packets) The tooltip explains that the policy will stop running when reaching either of the limits.

The third component is the **Rules Table**, which displays all **Match Traffic** rules configured for the policy. The default value is **Allow All Traffic**. Optionally, configure **Deny All Traffic**.

Figure 3-27: Allow All Traffic

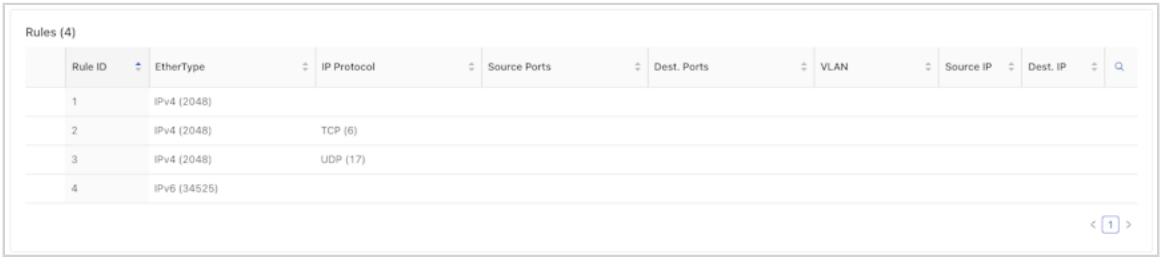


Figure 3-28: Deny All Traffic



When configuring custom rules, the **Rules Table** is displayed. The table is horizontally scrollable, and each column is searchable and sortable. The **Edit Policy** feature provides rule management, including **Edit**, **Add**, and **Delete** functionality.

Figure 3-29: Rules Table



The next component is the Interface Info Columns.

Figure 3-30: Information Columns

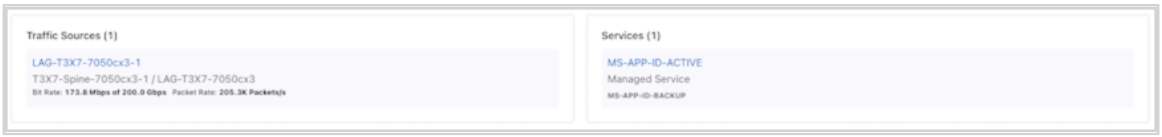


There are three primary columns: **Traffic Sources**, **Services**, and **Destination Tools**.

- The Traffic Sources column includes Filter Interfaces, vCenters, and CloudVision Portal associated with the policy.
- The Services column includes Managed Services and Services associated with the policy.
- The Destination Tools column includes Delivery interfaces and RN Fabric Interfaces associated with the policy.

These columns display the DMF Interface name in the **interface card**, and the name includes a link to the Interfaces page. The switch name and physical interface name appear in this format: **SWITCH-NAME / INTERFACE-NAME** under the DMF interface name. The bit rate and packet rate operational state data appear for each interface. Each column is only displayed if the policy has one or more interfaces of that type.

Figure 3-31: Traffic Sources



The services column renders for all policies that have at least one service. The service name appears for each card, which contains a link to either the Services or Managed Services page. Under the service name, the service type (Managed Service or Service) appears if the service has a backup name that also appears.

Figure 3-32: Managed Service



There is a special case for policies that have CloudVision port mirroring sessions. To differentiate the CloudVision source interfaces from the auto-generated DMF filter interfaces, DMF creates two columns: **CloudVision** and **Filter Interfaces**.

The cards in the CloudVision column show the connected CloudVision portal and the number of port mirroring sessions for each device in the CloudVision portal. Filter Interfaces and vCenters are now in the Filter Interfaces column. There are no differences between the Services and Destination Tools columns.

The last component only displays for policies with CloudVision port mirroring sessions.

Figure 3-33: Port Mirroring Sessions



The **Port Mirroring Session Entries** table shows all configured Port Mirroring Sessions for a CloudVision portal. The **Device**, **Source Interface**, **Monitor Type**, **Tunnel Source**, **Tunnel Endpoint**, **SPAN Interface**, and **Direction** columns display the same values configured in the Port Mirroring Table in the Add Traffic Sources component in the Create Policy flow. Each column is sortable.

For more information on the configuration flow for CloudVision port mirroring, please refer to the documentation in the **Create Policy** section.

Operational Details

Selecting the **Operational Details Tab** navigates to the Operational Details view.

Figure 3-34: Operational Details

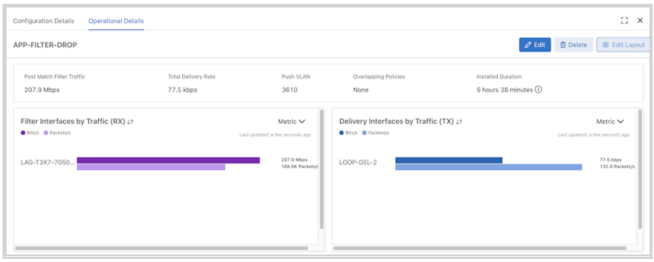
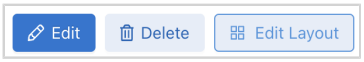
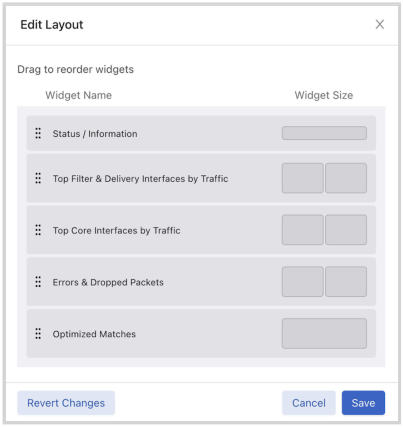


Figure 3-35: Action Buttons



- **Edit**: Selecting **Edit** opens the Editing Policy window for making changes to the policy.
- **Delete**: Selecting **Delete** deletes the policy.
- **Edit Layout**: Selecting **Edit Layout** opens the editing layout window. Move the widgets by dragging the components in order of user preference. Select **Save** to save the changes. DMF preserves the order of the widgets when the same user logs back in.

Figure 3-36: Edit Layout



Widgets

Status / Information

Status and information include basic operational information about the policy.

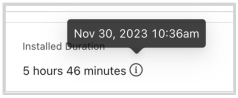
Figure 3-37: Operational Information



Installed Duration

Hover over the info icon to see the installed time in the UTC time zone.

Figure 3-38: Install Time



Top Filter and Delivery Interfaces by Traffic

Figure 3-39: Top Filter and Delivery Interfaces by Traffic

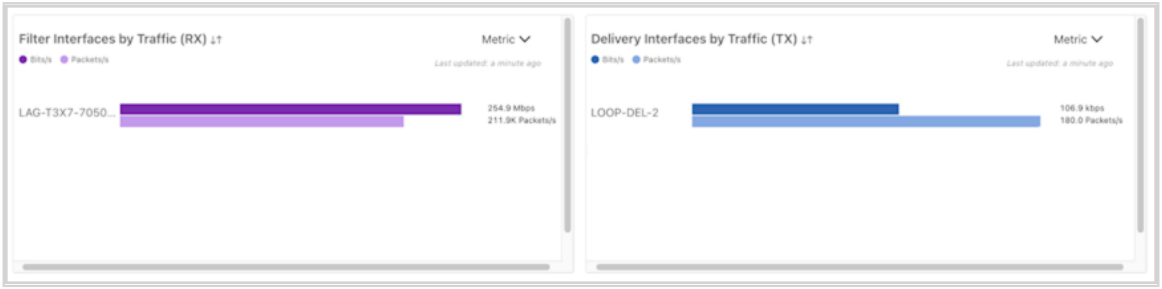
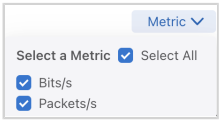


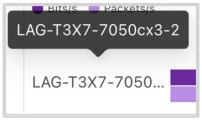
Figure 3-40: Select Metric



Select the **Metric Drop-down** menu and choose the metrics to display in the chart. Only the selected metrics appear in the **Badge**, **Labels**, and **Bar Chart**.

- **Badge**: Colored dots and text indicate the content represented by different bars in the bar chart.
- **Interface Name**

Figure 3-41: Labels

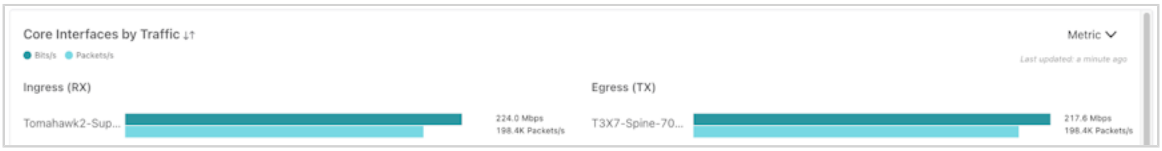


Hover over the **interface name** to see the full name in the tooltips.

- **Labels**: Display the number and unit corresponding to the bar.
- **Bar Chart**: Displays the numerical value of traffic.
- **Empty State**
  - Display title, last updated time, and disabled metric drop-down.
  - **Edit Policy** opens the edit policy window.

Top Core Interfaces by Traffic

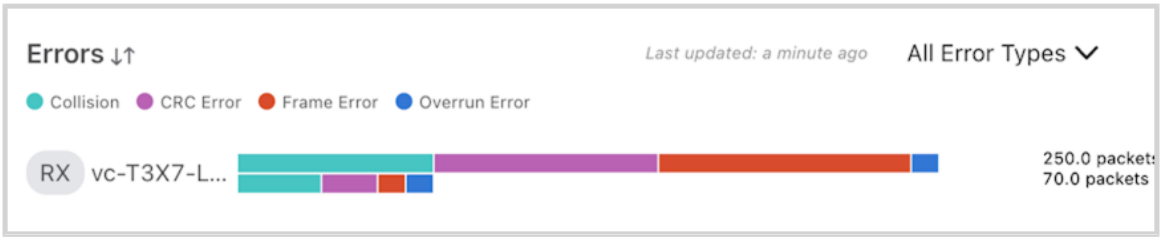
Figure 3-42: Top Core Interfaces by Traffic



The **Core Interfaces by Traffic** chart is similar to **Filter Interfaces** / **Delivery Interfaces by Traffic** charts, which have **Metric Drop-down**, **Badge**, **Interface Name**, **Labels**, and **Bar Charts** with similar functionality.

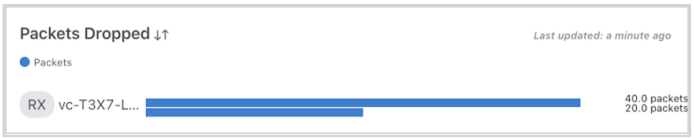
Errors & Dropped Packets

Figure 3-43: Errors



The **Errors** chart is similar to **Filter Interfaces** / **Delivery Interfaces by Traffic** charts, which have **Metric Drop-down**, **Badge**, **Interface Name**, **Labels**, and **Bar Charts** with similar functionality. Hovering over the bar displays all error counts and rate information.

Figure 3-44: Packets Dropped

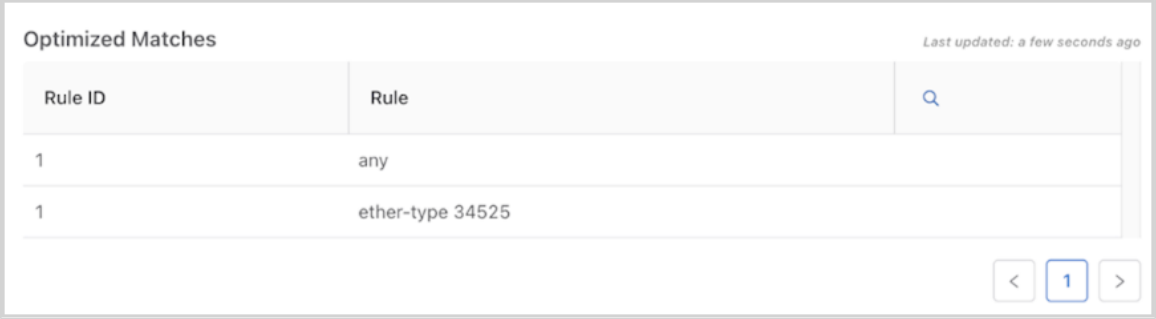


The **Dropped Packets** chart is similar to **Filter Interfaces** / **Delivery Interfaces by Traffic** charts, which have **Badge**, **Interface Name**, **Labels**, and **Bar Charts** with similar functionality. Hovering over the bar displays all packet dropped counts and rate information.

Optimized Matches

Displays optimized match rules.

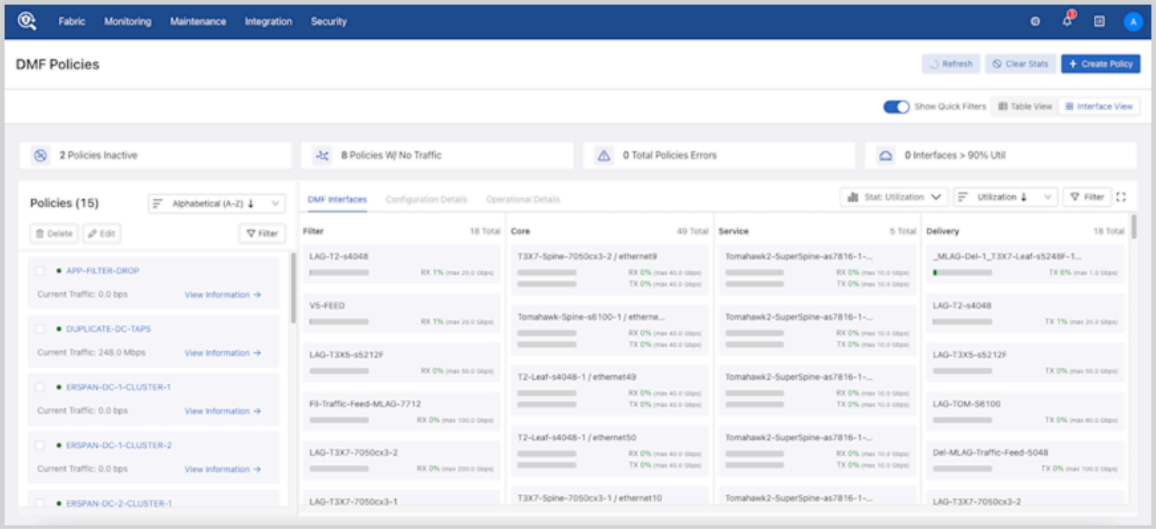
Figure 3-45: Optimized Matches



Interface View

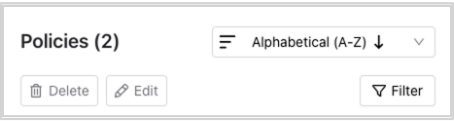
As a new feature of the **DMF Policies** page, the **Interface** view offers an alternative way to view policies, allowing for an intuitive visualization of all policies-related interfaces.

Figure 3-46: Interface View



Policies Column

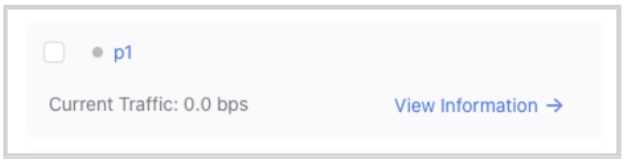
Figure 3-47: Policies Column



- A Policies header displaying count. The column shows the total count when no filters are applied, or the filtered policies count in the format of **x Associated**.
- The drop-down menu enables data sorting using multiple attributes.
- **Delete** deletes the selected policies.

- **Edit** opens the selected policy in edit mode.
- The **Filter** drop-down is similar to the table view filters but without an interface filtering option.

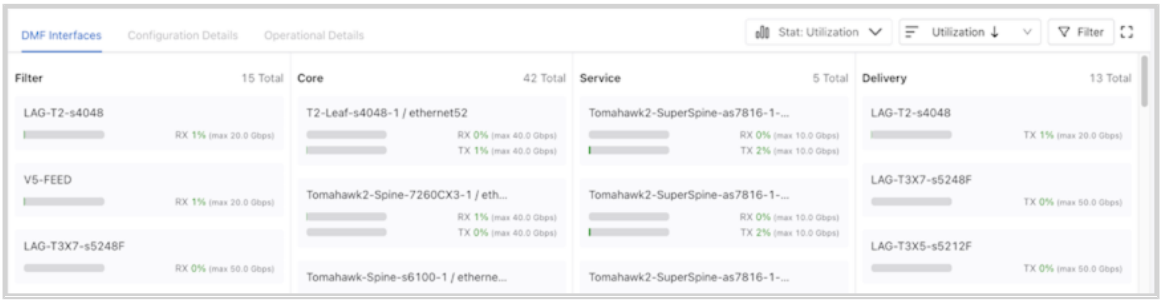
  Figure 3-48: Filter



- A list of policies with quick facts and user interactions.
- The **checkbox** enables policy selection for deletion and editing.
- Badges with different colors indicate policy run time status.
- Policy name with tooltip on hover displaying configuration, runtime, and detailed status.
- **Current Traffic** display in bps.
- Selecting **View Information** highlights the policy:
  - Only shows the interfaces associated with the selected policy in the **DMF Interfaces** tab.
  - Enable **Configuration Details** and **Operational Details**.
- Selecting an active policy card deselects the previously selected policy:
  - De-emphasizes the policy and resets card styles and tabs accessibility.
  - Reveals all the interfaces in **DMF Interfaces**.
  - Interface card highlights in the **DMF Interfaces** tab can co-exist, leading to a more granular search.

DMF Interfaces

Figure 3-49: DMF Interfaces



- Active tab by default
- **Header Row**
  - **Stat selector**: Choose between Utilization, Bit Rate, and Packet Rate to display in the subsequent interface info cards.

- **Sorter selector**: Choose between Utilization and interface name to sort the interfaces in ascending or descending order.
- **Filter drop-down**:
  - Utilization range filter
  - Switch name selector
  - DMF interface name selector

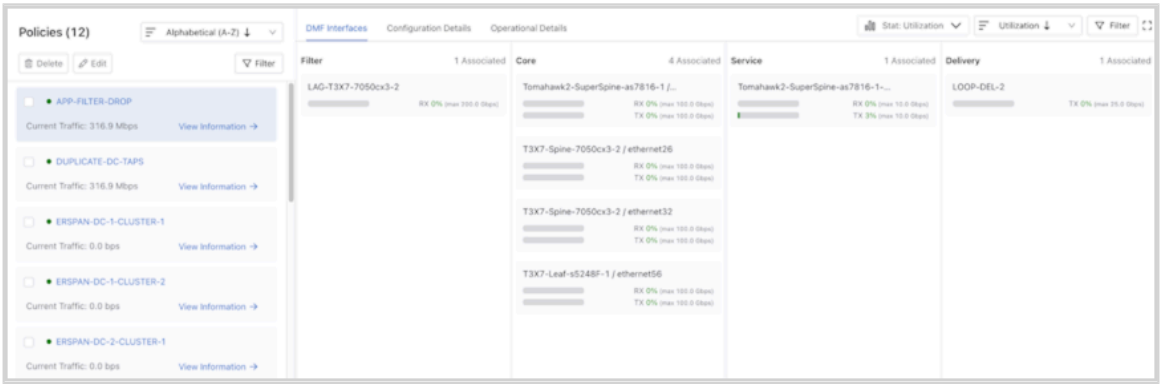
- **Interface Column**
  - **Header**: Specifies interface category and count, showing **X Associated** when filters apply and **X Total** otherwise.
  - Interface Information Card
    - Interface name
    - **Stat**
      - Utilization
      - Bit Rate
      - Packet Rate
    - **Text:** Display detailed information about the selected stat of the current interface.

Interaction

- Selecting one policy card:

  The selected policy card highlights and filters interfaces to only those configured to the policy and hides interfaces not configured in the selected policy.

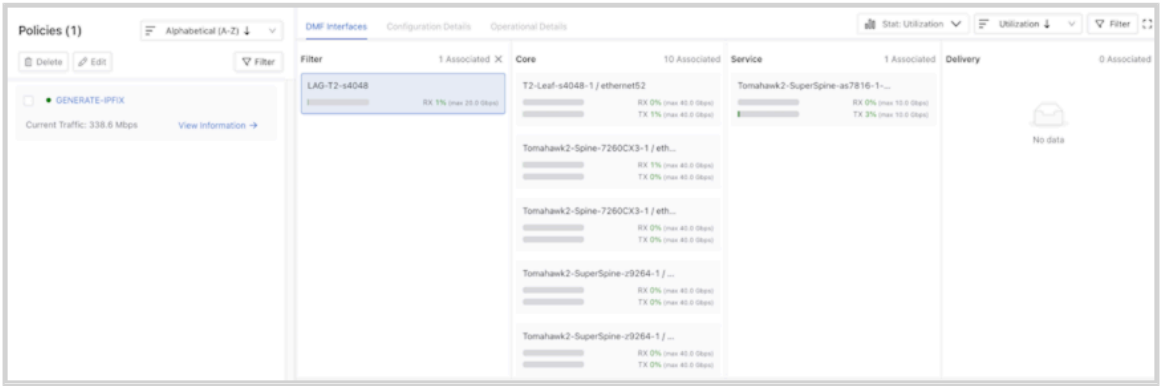
Figure 3-50: Policy Card



- Selecting one interface card:

The selected interface card highlights and filters policies to only those configured to the interface and hides interfaces not configured in the filtered policies mentioned above.

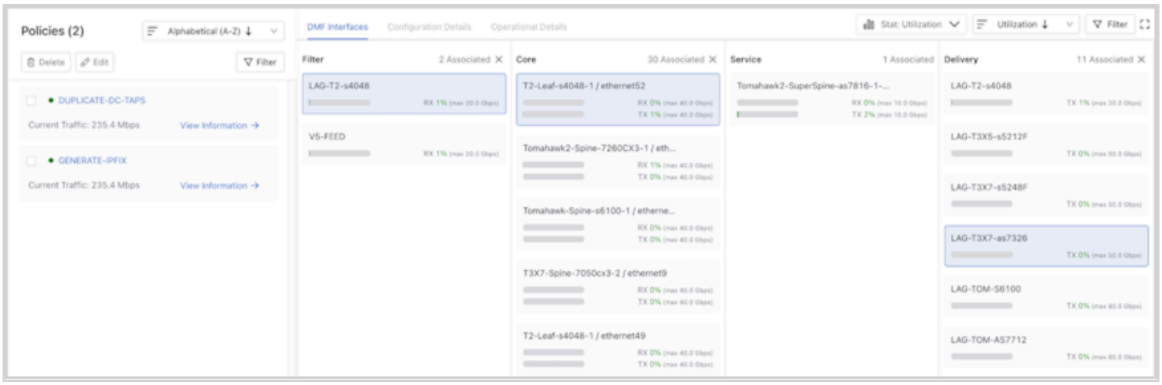
Figure 3-51: Single Interface Card



- Selecting multiple interface cards (any columns):

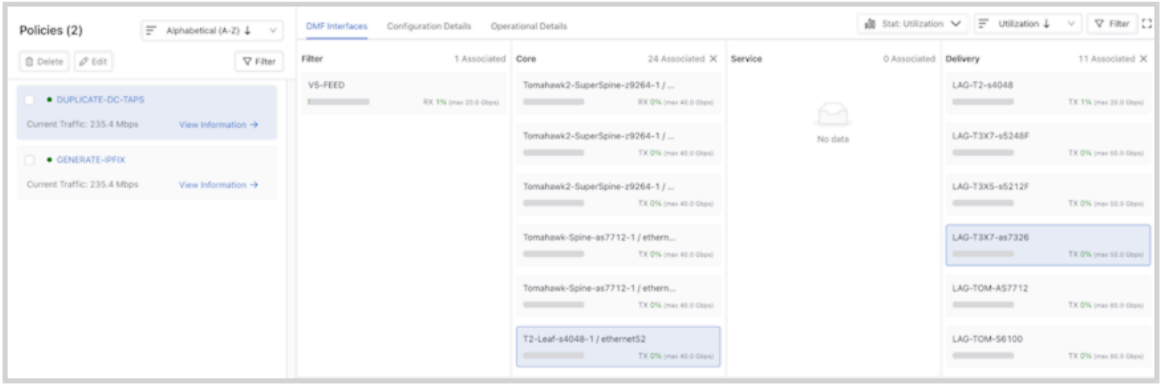
  The selected interface cards highlight and filter policies to only those configured on the selected interfaces and hide interfaces not configured in the filtered policies mentioned above.

Figure 3-52: Multiple Interface Cards



Highlighted policy and interface cards can co-exist, leading to a more granular search.

Figure 3-53: Policy and Interface Cards



Configuration Details

The GUI is similar to **Table View > Configuration Details**. Please refer to the Configuration Details section.

Operational Details

The GUI is similar to **Table View > Operational Details**. Please refer to the Operational Details section.

3.3 Policy Elements

Each policy includes the following configuration elements:

- **Filter interfaces**: these identify the ingress ports for analyzing the traffic for this policy. Choose individual filter interfaces or one or more filter interface groups. Select the **Select All Filter Interfaces** option, intended for small-scale deployments.
- **Delivery interfaces**: these identify the egress ports for analyzing the traffic as part of this policy. Choose individual delivery interfaces or one or more delivery interface groups. Like filter interfaces, a **Select All Delivery Interfaces** option is available for small deployments.
- **Action**: identifies the policy action applied to the inbound traffic. The following actions are available:
  - **Forward**: forwards matching traffic at filter ports to the delivery ports defined in a given policy. Select at least one or more filter and delivery interfaces.
  - **Drop**: drops matched traffic at the Filter ports. A policy with a drop action is often used in combination with another lower-priority policy to forward all traffic except the dropped traffic to tools. Use Drop to measure the bandwidth of matching traffic without forwarding it to a tool. Select at least one or more filter interfaces.
  - **Capture**: sends the selected traffic to a physical interface on the controller to be saved in a PCAP file. This option works only on a hardware Controller appliance. Select at least one or more filter interfaces. A policy with a capture action can only run for a short period. For continuous packet capture, use the DANZ Monitoring Fabric (DMF) recorder node. Refer to the chapter DMF Recorder Node for details.

    Note: The policy will not be installed if an action is not selected.

  - **Match rules**: used to select traffic. The selected traffic is treated based on the action, with the most common action being **Forward**, i.e., forward-matched traffic to delivery interfaces. If a match rule is not specified or the match rule is **Deny All Traffic**, the policy is not installed. One policy can specify multiple match rules, differentiating each rule by its rule number.

    Note: The rule numbers do not define the order in which the rules will be installed or processed. The numbering allows a user to list them in order.

  - **Managed services** (optional): identifies additional operations to perform, such as packet slicing, time stamping, packet deduplication, packet obfuscation, etc., before sending the traffic to the selected delivery interfaces.
  - **Status** (optional): enables or disables the policy using the **`active`** or **`inactive`** sub-command from the config-policy sub-model. By default, a policy is active when initially configured.
  - **Priority** (optional): unless a user specifies, all policies have a priority of ***100***. When sharing filter/ ingress ports across policies, a policy with a higher priority will get access to matching traffic first. Traffic not matched by the policies with the higher priority then gets processed according to policies with lower priority. Overlapping policies are also not created when two policies have different priorities defined.
  - **Push VLAN** (optional): when a user configures the Auto VLAN Mode push as push-per-policy (i.e., to **Push Unique VLAN on Policies**, every policy configured on DMF gets a unique VLAN ID. Typically, this VLAN ID is in the range of ***1-4094*** and auto-increments by ***1***. However, to specific policy with a

specific VLAN ID, first define a smaller VLAN range using the command **`auto-vlan-range`** and then pick a VLAN outside that range to attach to a specific policy. This attachment of a specific VLAN to a specific policy can be done in the CLI using the CLI command **`push-vlan`** or in the GUI by selecting **Push VLAN** from the **Advanced Options** drop-down and then specifying the VLAN ID.

- **Root switch** (optional): when a core switch (or core link) goes down, existing policies using that switch are rerouted using other core switches. When that switch comes back, the policy does not move back. In some cases, this causes traffic overload. One way to overcome this problem is to specify a root switch in each policy. The policy is rerouted through other switches when the root switch goes down. When the root switch comes back, DMF reroutes the policy through the root switch again.

Policies can include multiple filter and delivery interfaces, and services are optional. Traffic that matches the rules in any policy affiliated with a filter interface forwards to all the delivery interfaces defined in the policy.

Except for a capture action policy, a policy runs indefinitely once activated. Optionally schedule the policy by specifying a starting time and period for which the policy should run and specify the number of received packets in the tool, after which the policy automatically deactivates.

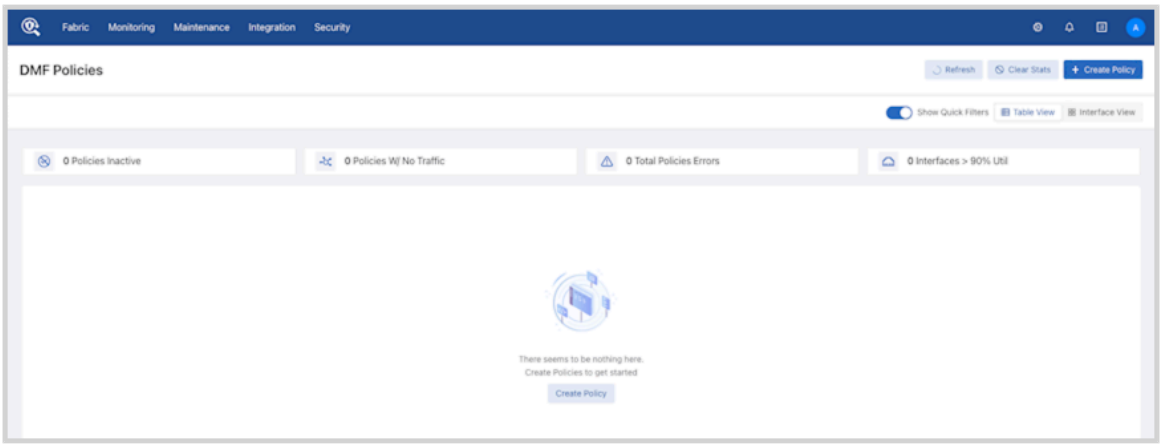
Note:

1. Create and configure all interfaces and service definitions before creating a policy that uses them.
2. Use only existing interfaces and service definitions when creating a policy. When creating a policy with interfaces or service definitions that do not exist, the policy may enter an inconsistent state.
3. If this happens, delete the policy, create the interfaces and service definitions, and then recreate the policy.

3.4 Configure a Policy

There are two possible entry points for creating a policy. The first is via **Create Policy**, continuously displayed on the top-right corner of the **DMF Policies** page, or the second is via **Create Policy**, which appears on the central panel of the same page when no configured policies exist.

Figure 3-54: DMF Policies



Selecting **Create Policy** opens the new **Policy Creation** configuration page, which supports moving, minimizing, expanding, collapsing, and closing the window using the respective icons in the menu bar.

Figure 3-55: Create Policy

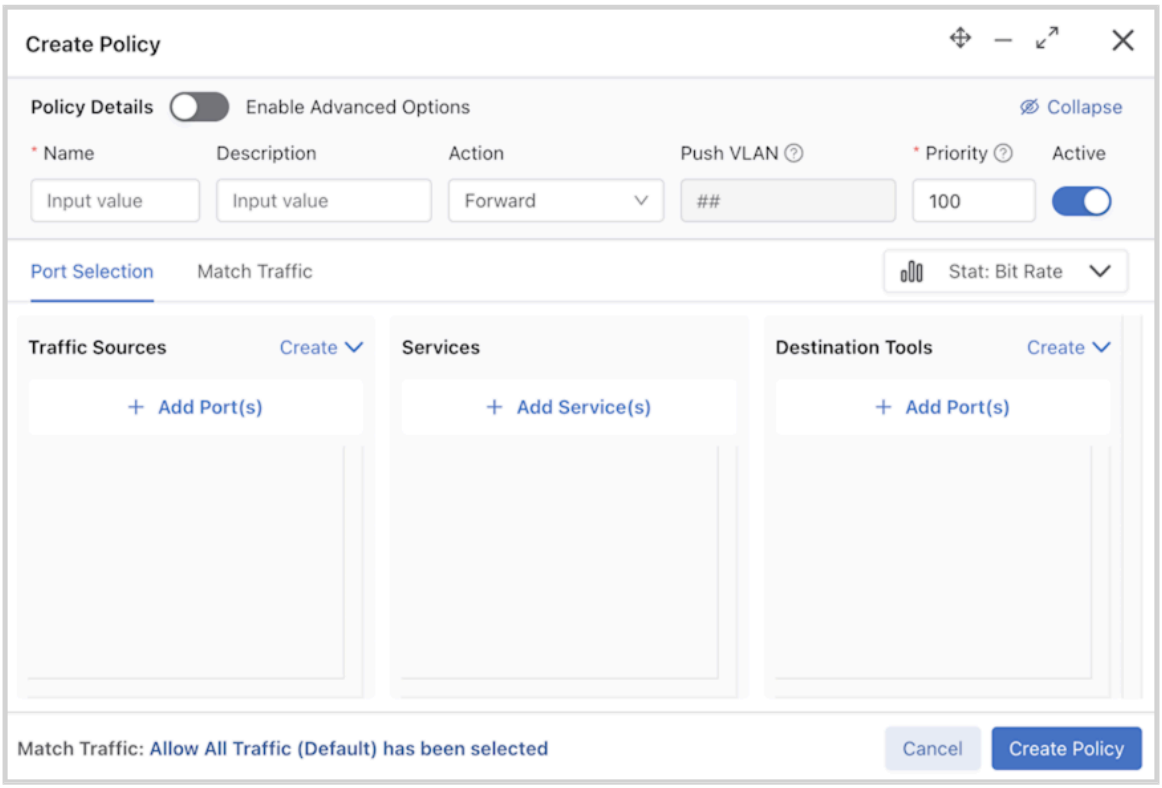


Figure 3-56: UI Controls



Move: Select (and hold) any part of the title section of the window or the icon to drag and reposition as required. Moving the window in full-size mode is not possible.

Expand: Use the icon to enlarge the window.

Minimize: Use the icon to minimize the window and the icon to return to the standard view.

Proceed to the following sections for create and manage policies.

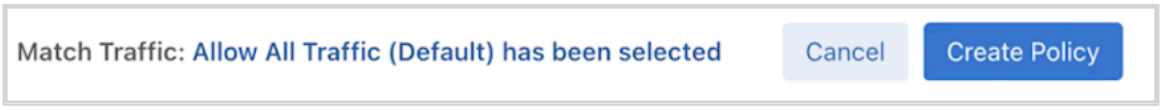
3.4.1 Create a New Policy

Create a New Policy

To create a new Policy, complete the required fields in the **Policy Details** section and configure settings under the **Port Selection** tab (optional) and the **Match Traffic** tab (optional). Please refer to the Policy Details, Port Selection Tab, and Match Traffic Tab sections for more detailed information on configuring settings.

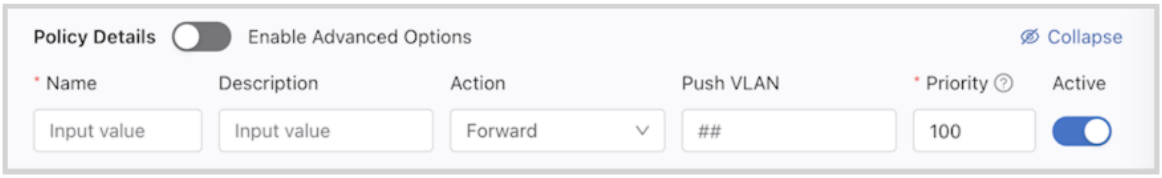
Once configured, select **Create Policy** to save the changes and finish the policy creation.

Figure 3-57: Create Policy



Policy Details

Figure 3-58: Policy Details



Enter the primary information for the policy:

- **Policy Name** (must be unique)
- **Description**
- **Policy Action**: Capture, Drop, Forward (default)

> **Note:** The Destination Tools column is not available when Drop and Capture actions are selected.

- **Push VLAN**
- **Priority**: By default, set to 100 if no value is specified.
- **Active**: By default, set to enabled.
- **Advanced Options**: By default, disabled.

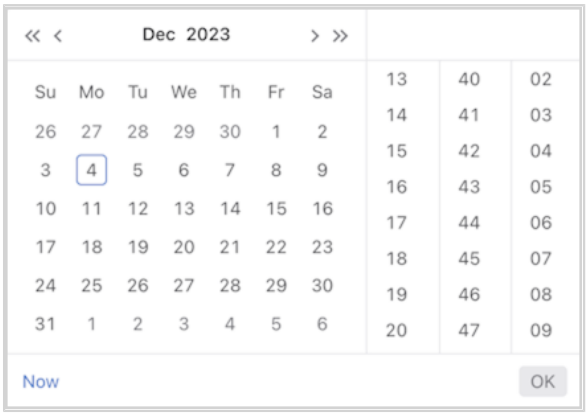
When **Advanced Options** is enabled, the following configuration settings are available:

Figure 3-59: Advanced Options



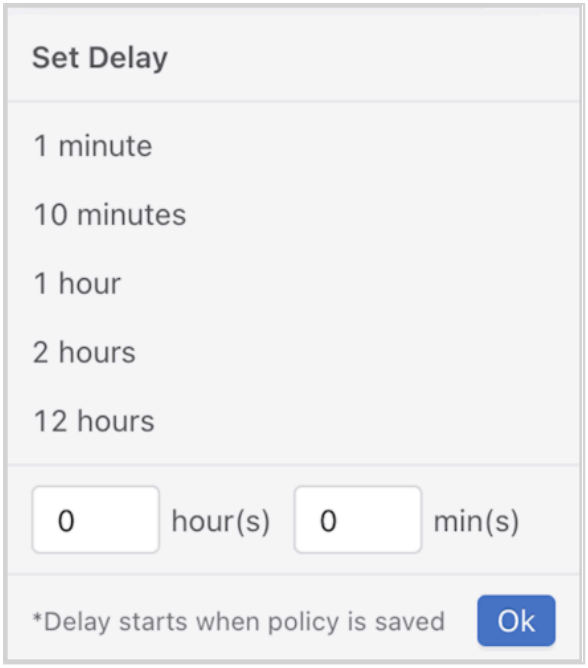
- **Scheduling**: There are four options:
  - **Automatic**: The policy runs indefinitely.
  - **Now**: The policy starts running immediately; use **Run Time** to determine when the policy should stop.
  - **Set Time**: Set a specific date and time to start the policy.

    Figure 3-60: Scheduling



- **Set Delay**: Start the policy using relative time options.

  Figure 3-61: Set Delay



- **Run Time**: There are two options:
  - **Always**: (default).
  - **For Duration**: Selecting **For Duration** allows using **Time Input** to set the time number and the **Unit** selector to set the time unit. Select the checkbox to use **Packet Input** and enter the required packet number (1000, by default).

Figure 3-62: Run Time



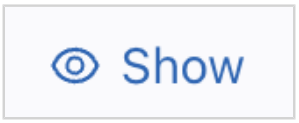
- **PTP Timestamping**: Disabled by default.
- **Root Switch**: By default, set to a locked state. Select the lock icon to unlock and select a root switch.

Additional Controls

Figure 3-63: Collapse



Figure 3-64: Show

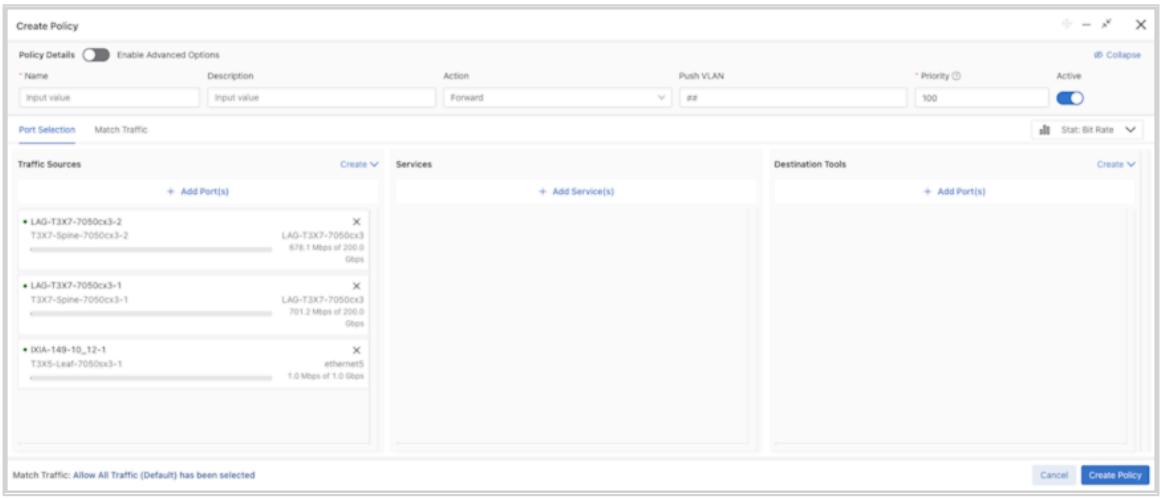


- **Collapse and Show:** Visually hide or unhide the basic policy configurations to manage the view of the other configuration fields.

3.4.2 Traffic Sources

The **Traffic Sources** column displays the associated traffic sources in the policy.

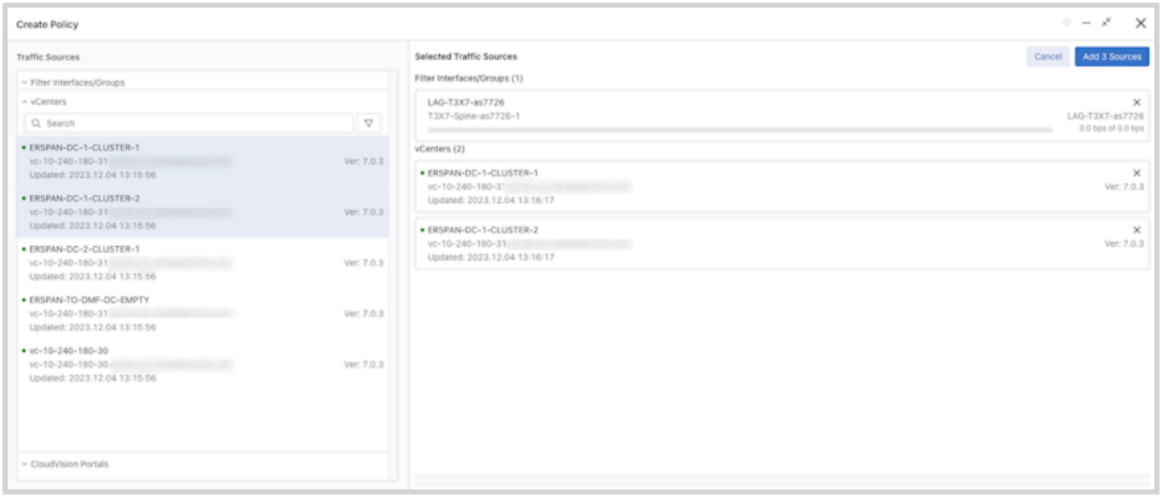
Figure 3-65: Traffic Sources



To add Sources, select **Add Port(s)**. The page allows adding **Filter Interfaces** and **Filter Interface Groups**, **vCenters**, **NSX**, or **CloudVision Portals**.

> **Note:** The left column has three multiple groups. Select the corresponding type of traffic source to view the available selections. After making all desired selections, confirm them using **Add N Sources**.

Figure 3-66: Add Sources

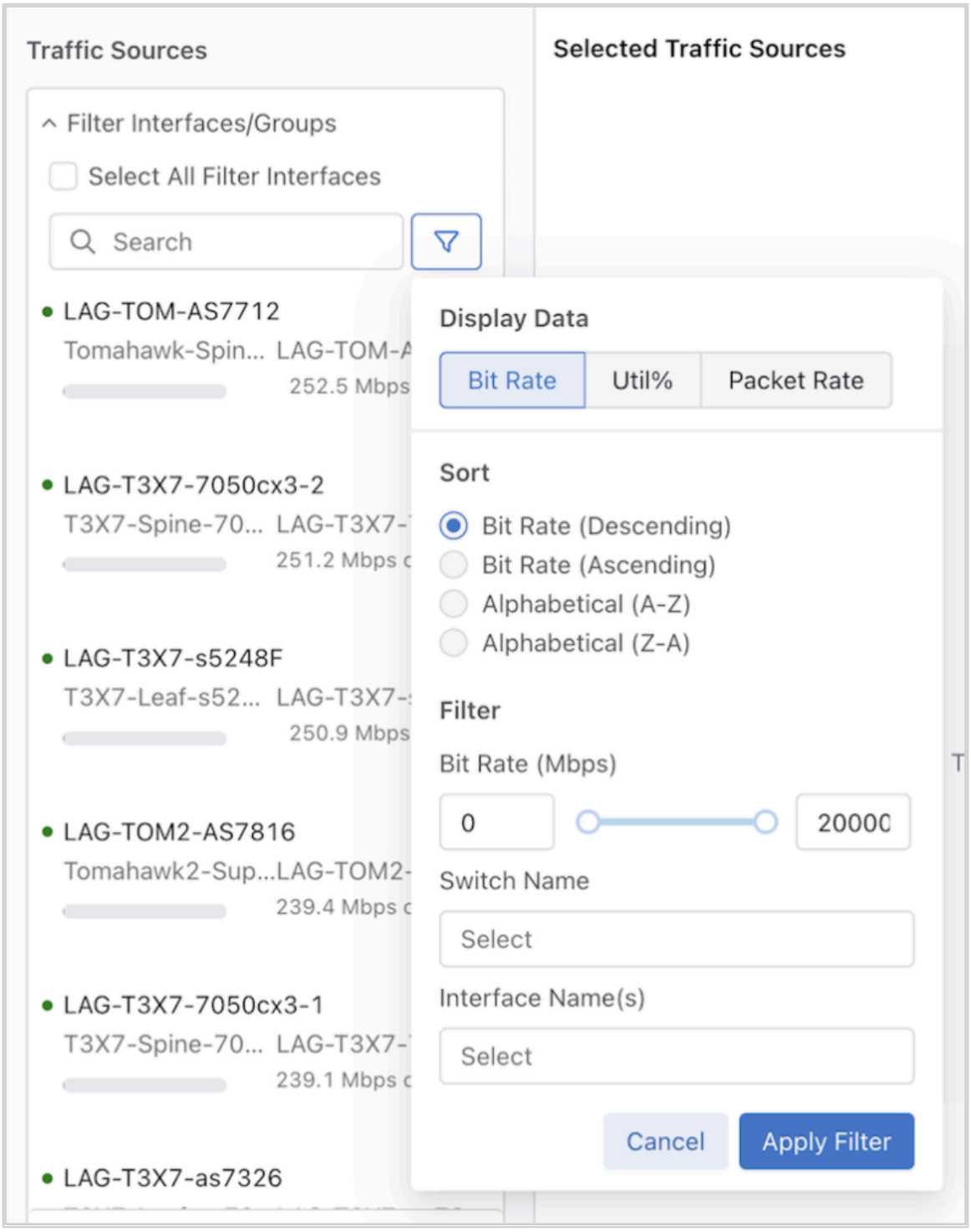


Search for interfaces using the search bar and the available information in the interface tiles. Selecting the icon reveals sorting and filtering options using **Display Data**, which includes:

- **Sort** - By default, DMF sorts the data in descending Bit Rate order. Optionally, sort the data by ascending Bit Rate order or alphabetically.

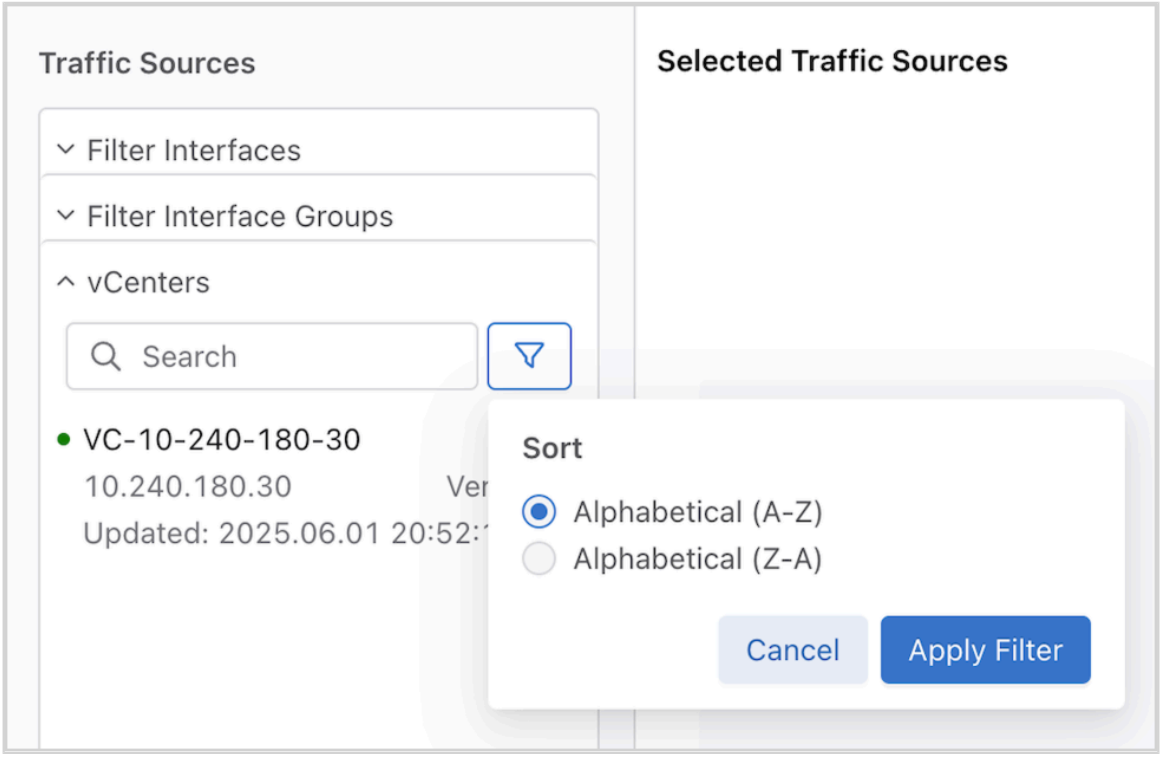
- **Bit Rate** (default), **Utilization** percentage, or **Packet Rate**
- **Switch Name**
- **Interface Name(s)**

Figure 3-67: Traffic Sources Display Data



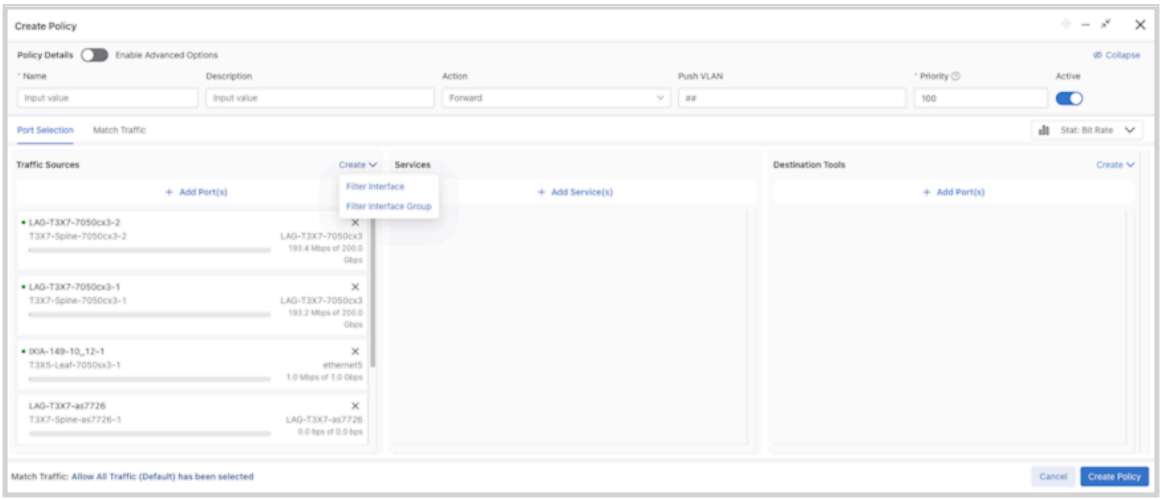
DMF sorts Interface Groups, vCenters, NSX, and CloudVision Portals alphabetically (A-Z, by default).

Figure 3-68: Sort Traffic Sources



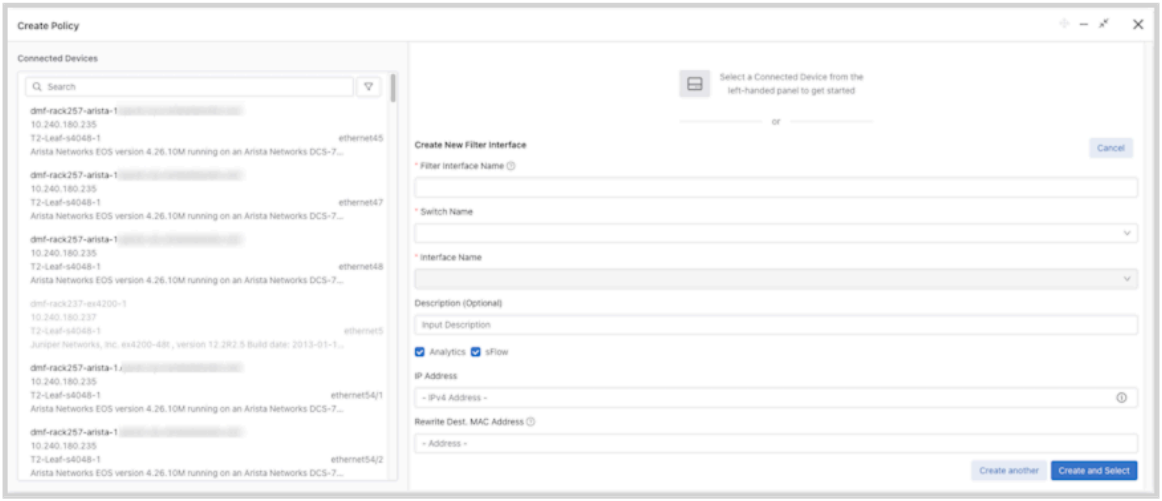
When a Filter Interface has not been created yet, **Create** has two selections: **Create Filter Interfaces** and **Filter Interface Groups**.

Figure 3-69: Filter Interfaces / Filter Interface Groups



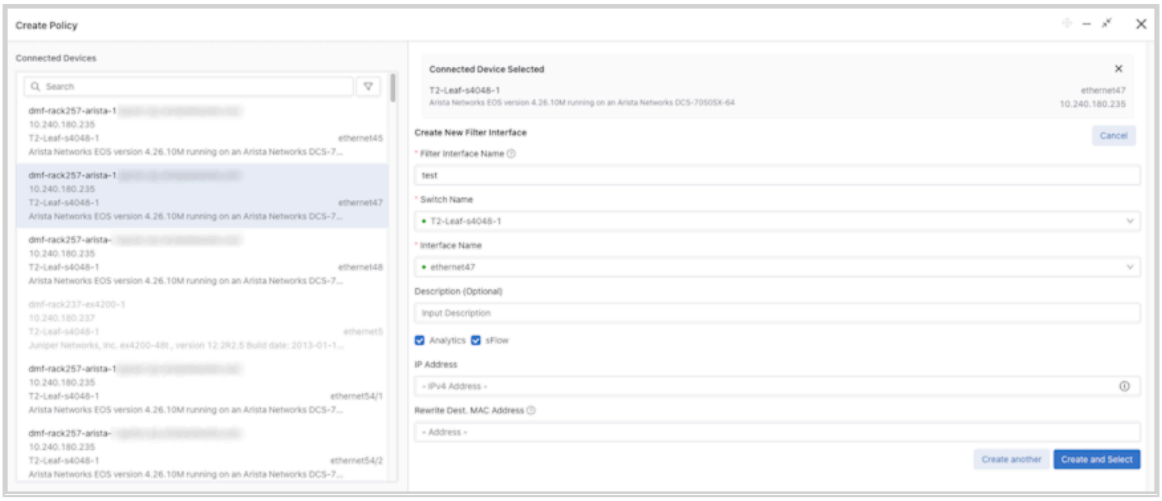
Selecting **Create Filter Interface** opens a form to configure a **Filter Interface**. Enter the required settings to configure the new **Filter Interface**.

Figure 3-70: Configure Filter Interface



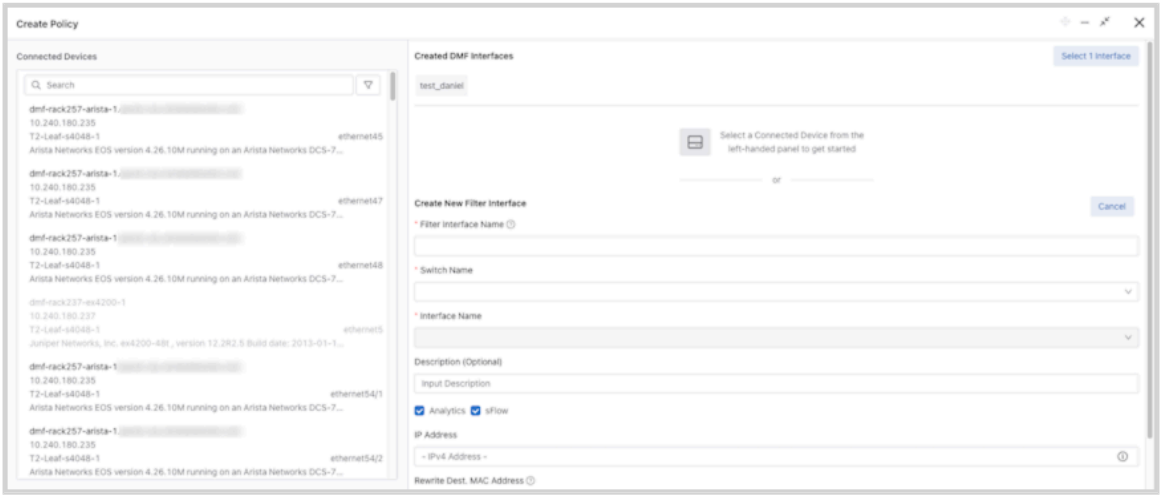
Alternatively, the left column allows the selection of an existing connected device to pre-populate the **Switch Name** and **Interface Name** fields and to configure a **Filter Interface** based on a connected device. Select **Create and Select** to create the **Filter Interface** and associate it with the current policy.

Figure 3-71: Associate Filter Interface



To create multiple **Filter Interface(s)**, select **Create another** to create an interface using the current configuration. This action clears the form to allow the creation of an additional **Filter Interface**.

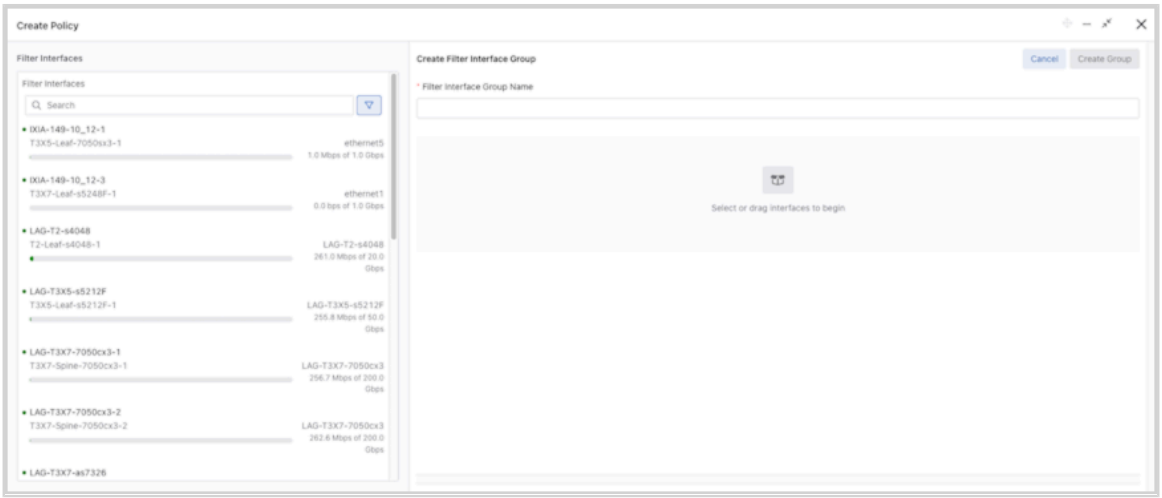
Figure 3-72: Add Multiple Filter Interfaces



Note: Select (n) interface associates all created **Filter Interfaces** to the current policy.

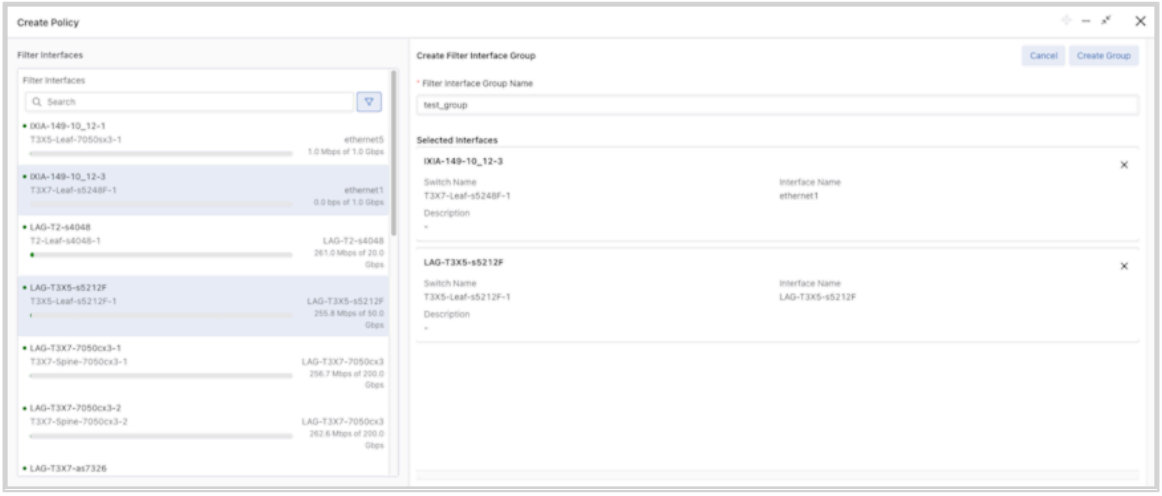
Select **Create Filter Interface Group** to create a group of filter interfaces.

Figure 3-73: Create Filter Interface Group



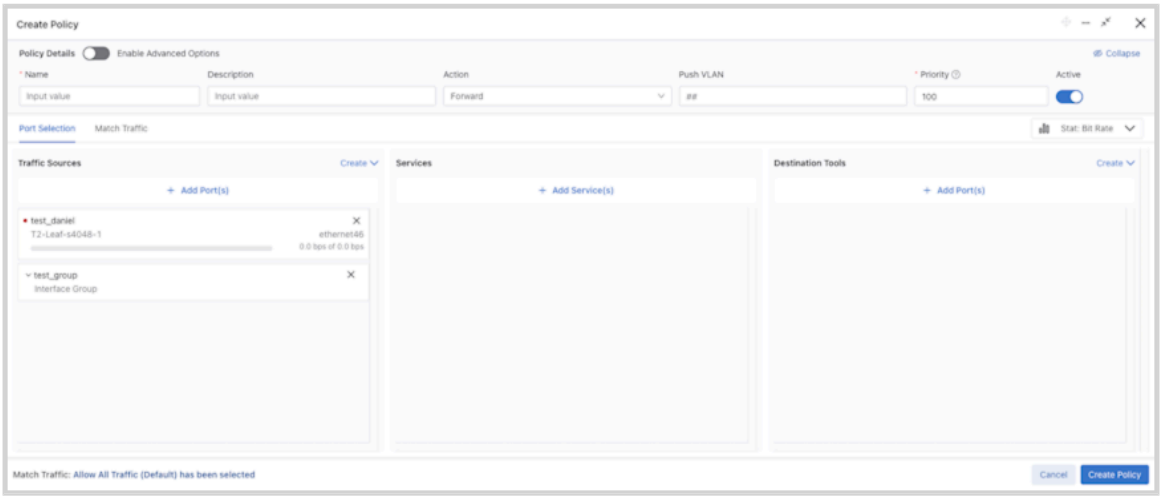
Select one or more filter interfaces to create a **Filter Interface Group**.

Figure 3-74: Add Filter Interfaces



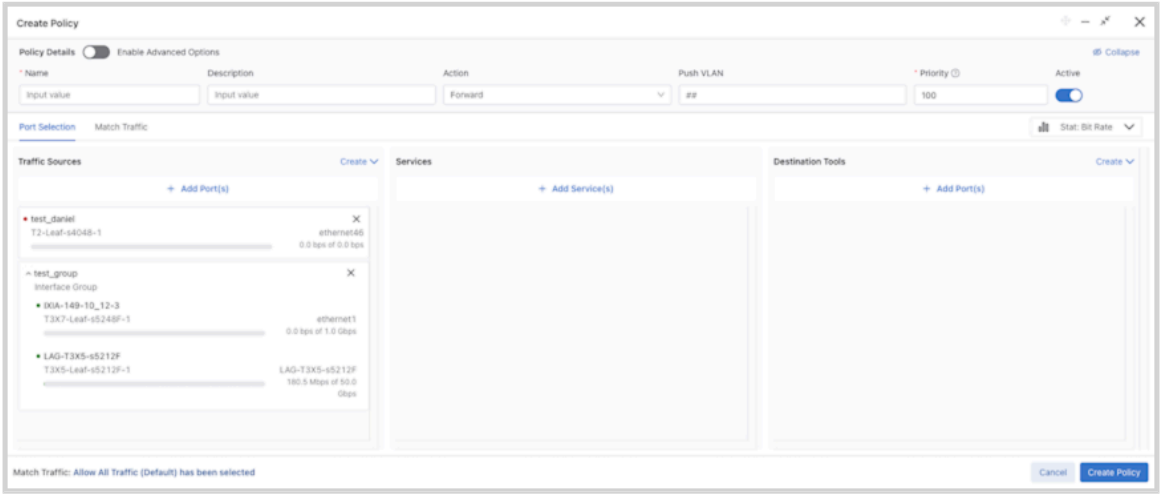
Select **Create Group** to create the **Filter Interface Group** and associate the group with the current policy.

Figure 3-75: Create Group



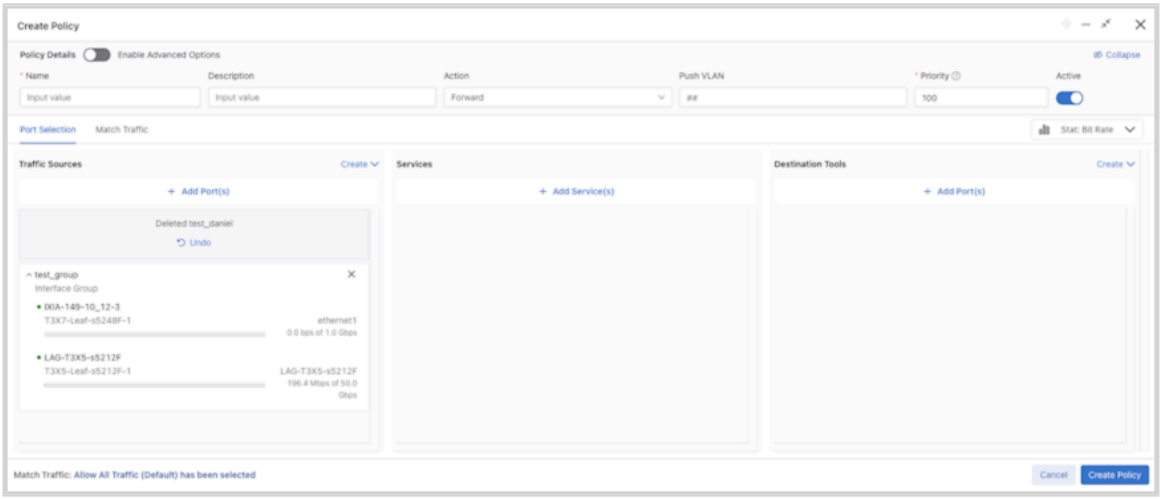
Expand the group tile to view interfaces within an **Interface Group**.

Figure 3-76: Expand Details



Note: Selecting the **x** icon on the top right of each tile disassociates the Filter Interface from the current policy. Selecting **Undo** restores the association.

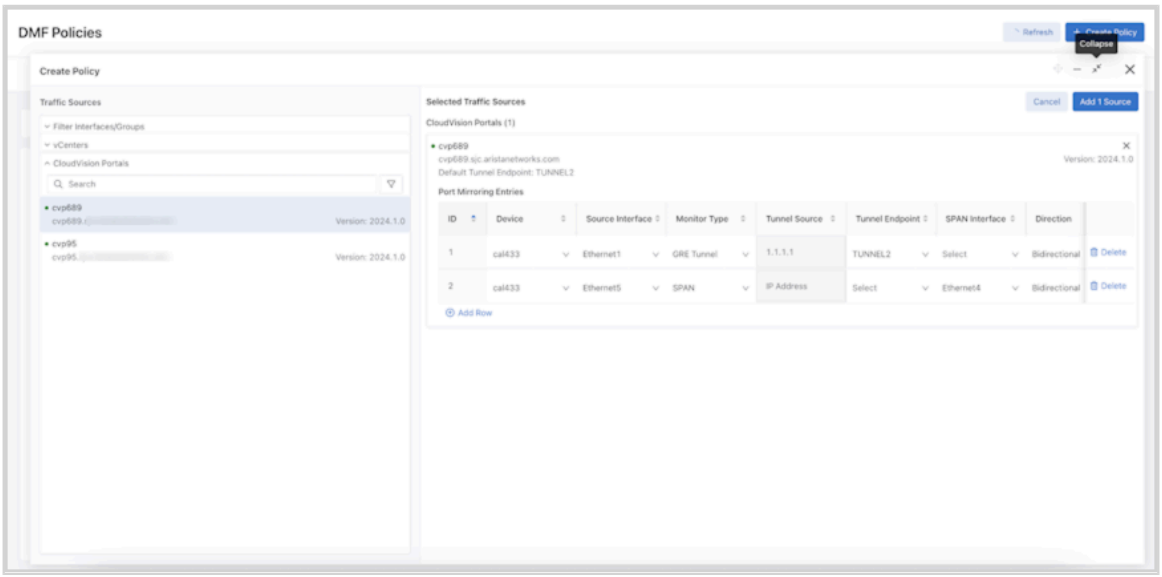
Figure 3-77: Disassociate Filter Interface



CloudVision Portals

The **Create Policy** window lists **CloudVision Portals** connected to DMF and includes the **CloudVision Portal** name, the portal hostname, and the current software version. Select a card to add a CloudVision **Port Mirroring Table**. The card displays similar information and the default Tunnel Endpoint.

Figure 3-78: CloudVision Portals



An empty port mirroring table initializes to add rows to the table for configuring port mirroring sessions.

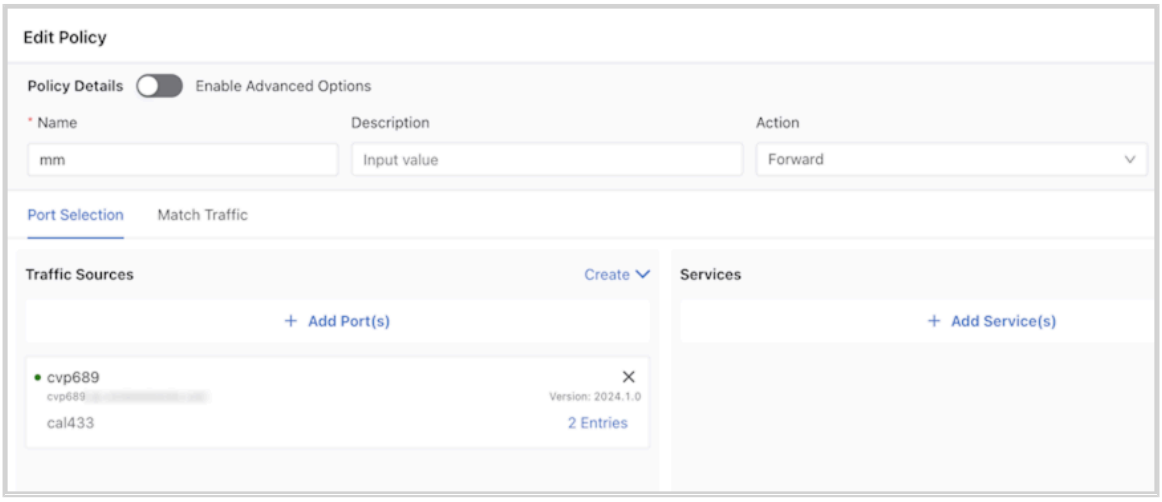
Use the following guidelines to configure a port mirroring session:

- Each row must contain a **Device** and Source **Interface**. This interface in the CloudVision production network will mirror traffic to DMF.
- Each interface must select a **Monitor Type**: **GRE Tunnel** or **SPAN**.

  Note: SPAN requires a physical connection from the CloudVision Portal to DMF. The default value for **Tunnel Endpoint** is the CloudVision Portal's Default Tunnel Endpoint.

- Each device must have the same **Tunnel Endpoint** and **Tunnel Source** values across the policies. Each interface on a device must have an identical destination configuration (**GRE Tunnel**, **GRE Tunnel Source**, and **SPAN Interface**) across the policies.
- The default traffic direction is **Bidirectional** but configurable to **Ingress** or **Egress**.
- After configuring the **Port Mirroring Table**, select **Add Sources** to return to the Main Page of the **Create Policy** configuration page.

Figure 3-79: Edit Policy

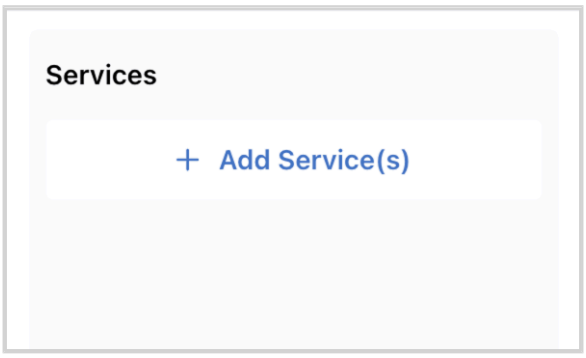


After configuring **Port Mirroring**, the card appears in the **Traffic Sources** section. To edit the **Port Mirroring Table**, select the **X Entries** link.

3.4.3 Services

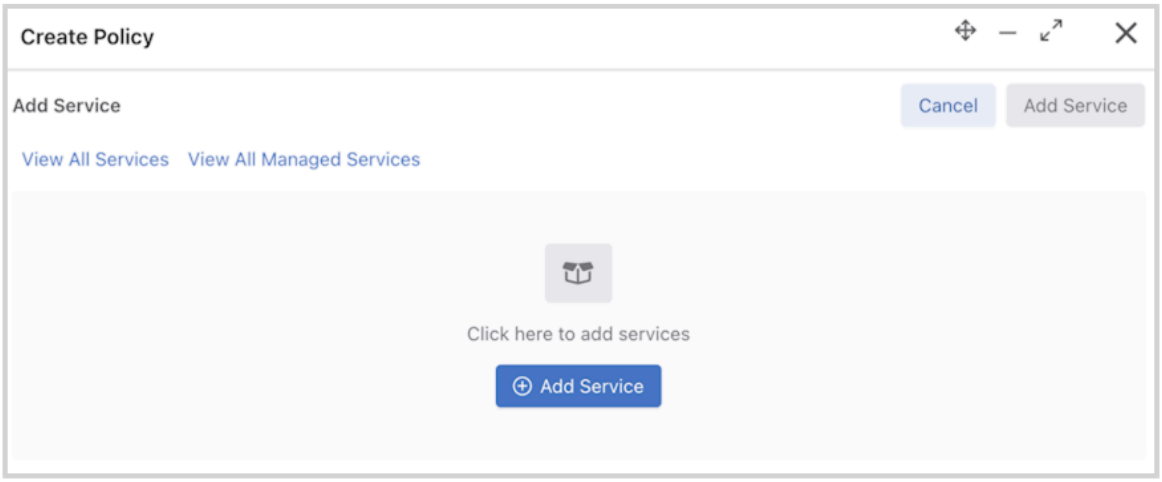
The **Services** column displays the **Services** and **Managed Services** associated with the policy. **Add Service(s)** opens a new page to specify additional services.

Figure 3-80: Services Add Services



View All Services and **View All Managed Services** open the DMF **Services** and **Managed Services** pages, respectively. **Add Service** opens a configuration panel to specify **Service** information. If there are **Services** associated with this policy, they are listed and available to edit.

Figure 3-81: View All Services / View All Managed Services

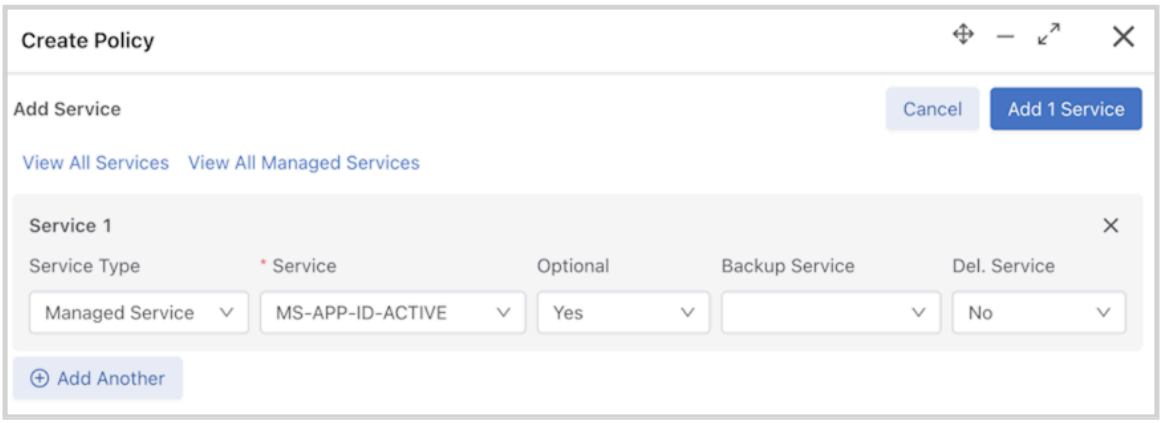


For each **Service**, specify:

- **Service Type**: Managed or Unmanaged.
- **Service**: Name of the Service (required).
- **Optional**: Whether the Service is optional.
- **Backup Service**: Name of the backup Service.
- **Del. Service**: If the Managed Service type is selected, whether to use it as a Delivery Service.

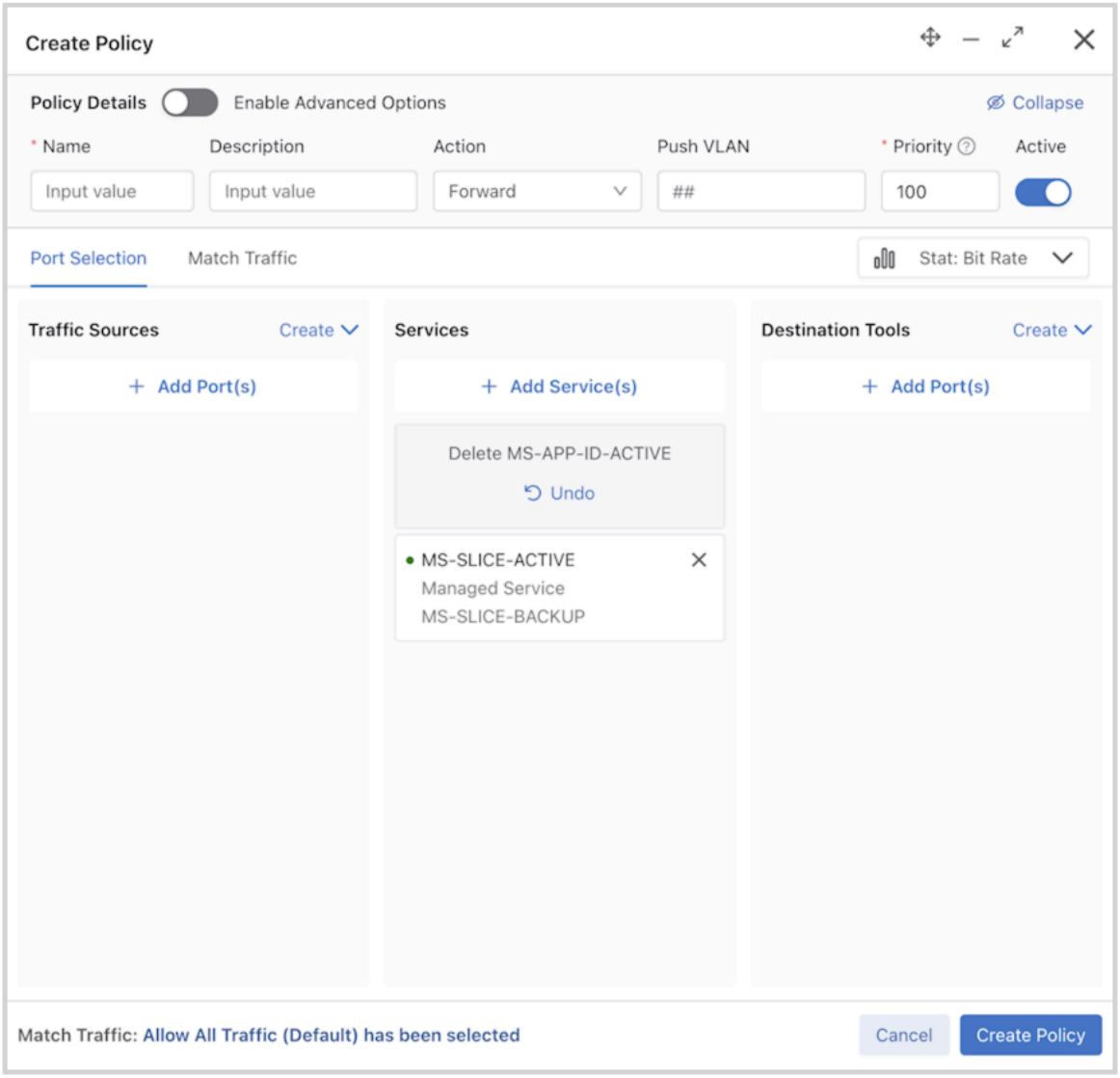
Select **Add Another** to populate a new row to add another **Service**. **Add (n) Services** associates the Services with the policy.

Figure 3-82: Add Another Service



After adding the services, they appear in the **Services** column. Select the **x** icon on the **Service** tile to disassociate the **Service** from the policy. While remaining on the page, if required, re-associate the Service by selecting **Undo**.

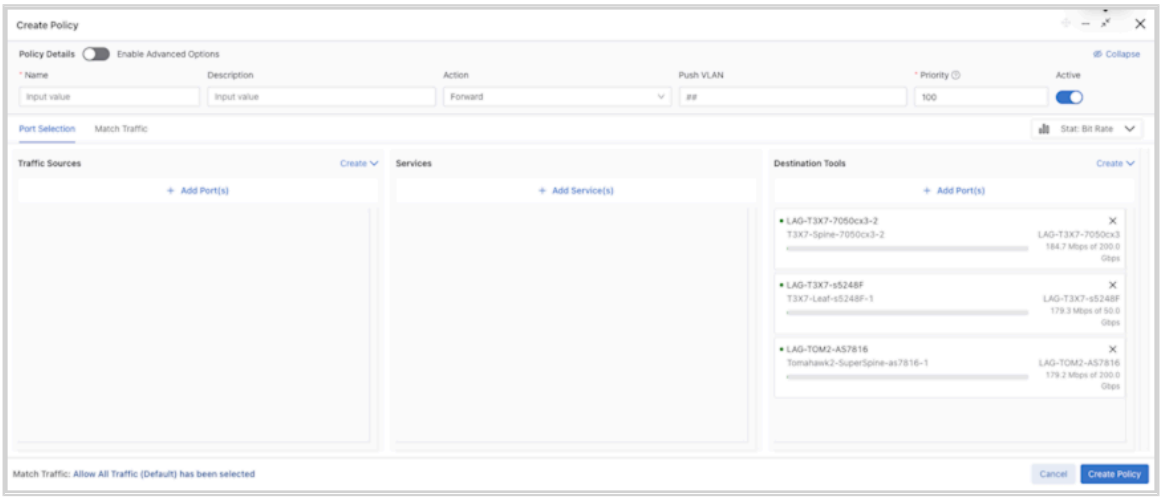
Figure 3-83: Service Added



3.4.4 Destination Tools

The **Destination Tools** column displays the associated Destination Tool ports to a given policy.

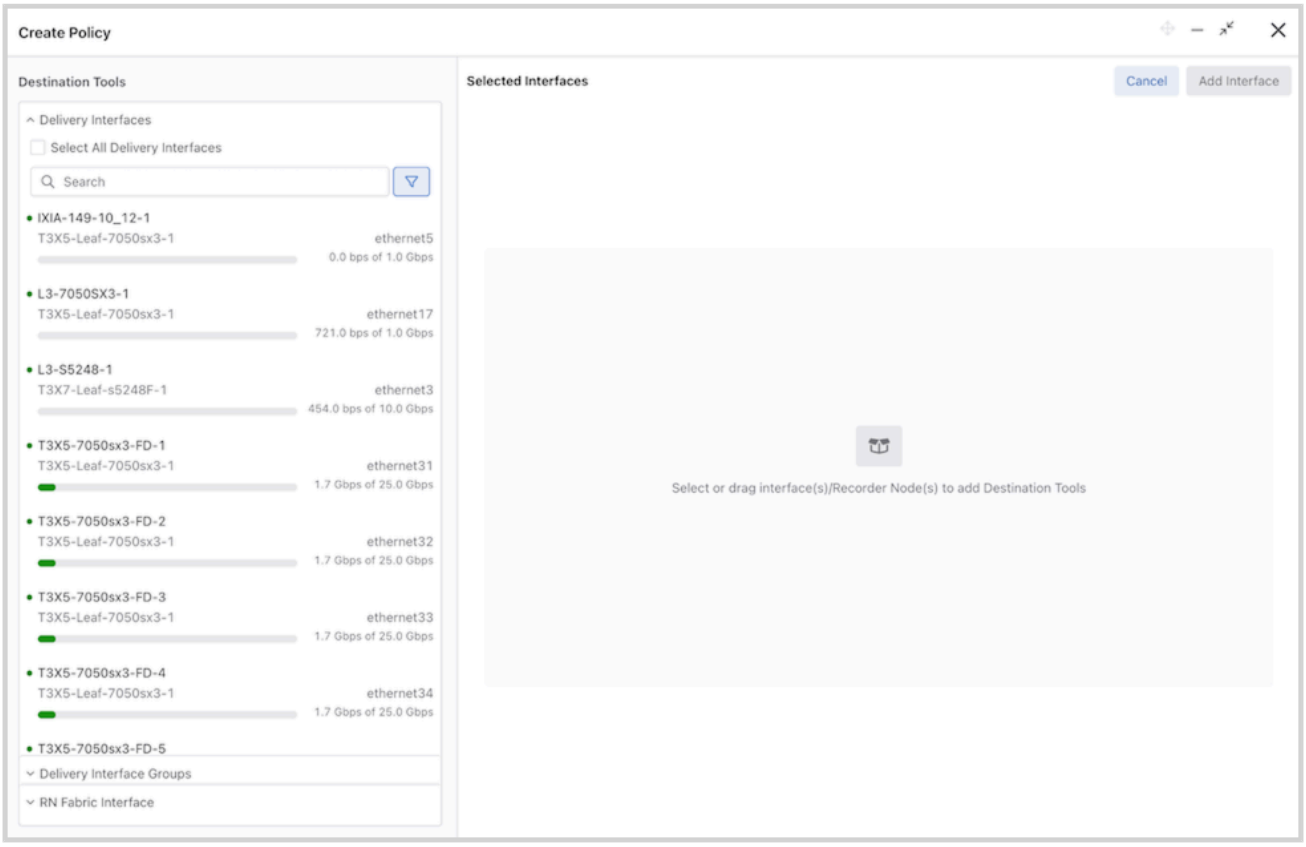
Figure 3-84: Destination Tools



Use **Add Port(s)** to add more destinations. The configuration page allows adding **Delivery Interfaces**, **Delivery Interface Groups**, or **Recorder Node Fabric Interfaces**.

Note: The left column has two multiple groups. Select the corresponding type of **Destination Tools** to see the available selections. After making the desired selections, confirm using **Add (n) Interfaces**.

Figure 3-85: Add Interfaces



Interfaces can be searched by the available information in the interface tiles using the search bar. Selecting the icon reveals sorting and filtering options using **Display Data**, which includes:

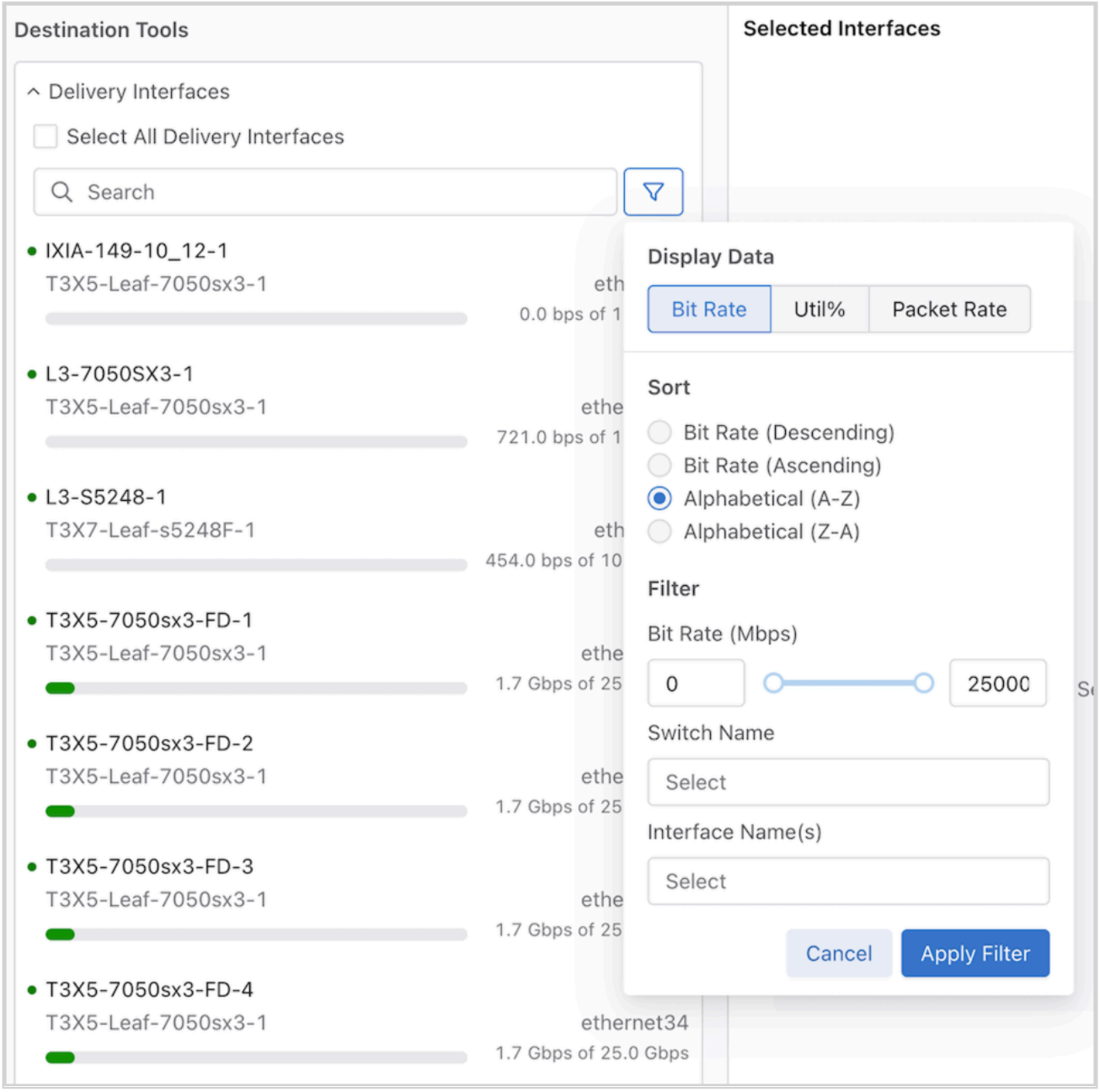
Sort - By default, DMF sorts the data in descending Bit Rate order. Optionally, sort the data by ascending Bit Rate order or alphabetically.

Bit Rate (default), **Utilization** percentage, or **Packet Rate**

Switch Name

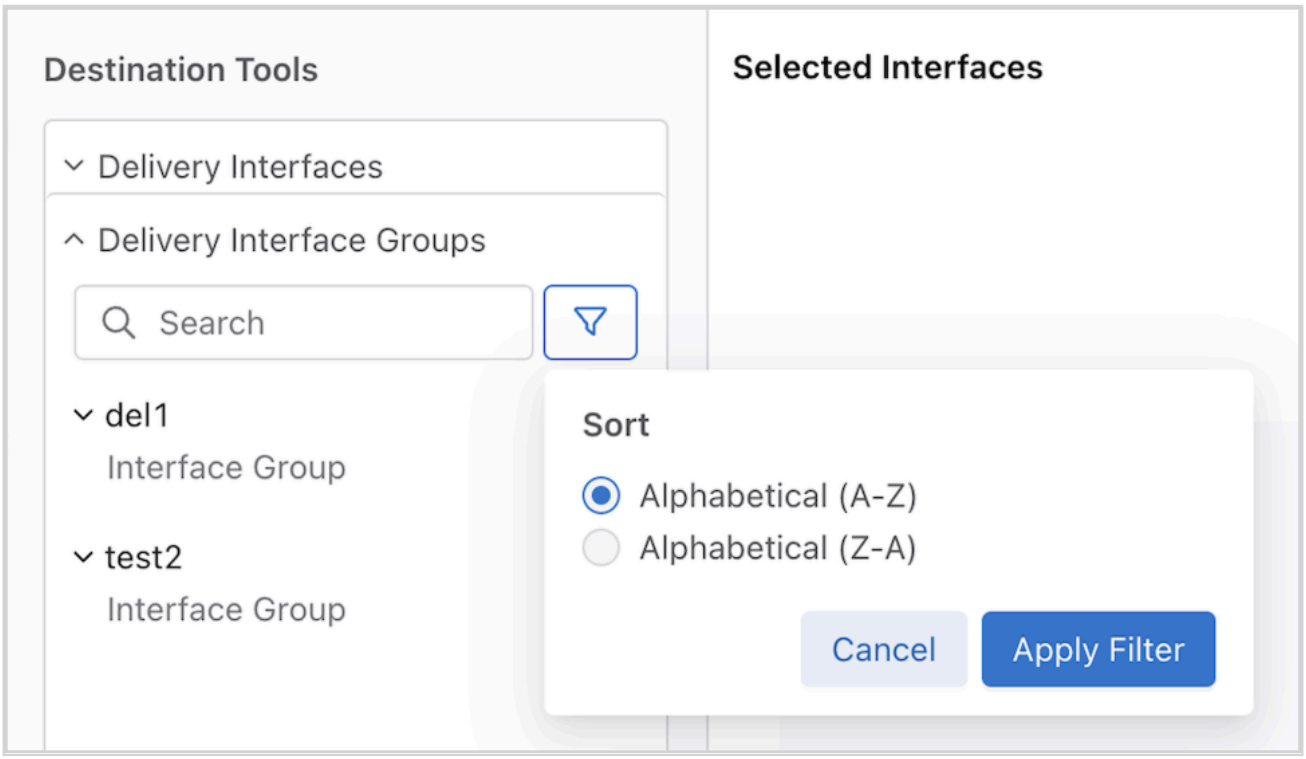
Interface Name(s)

Figure 3-86: Filter Destination Tools



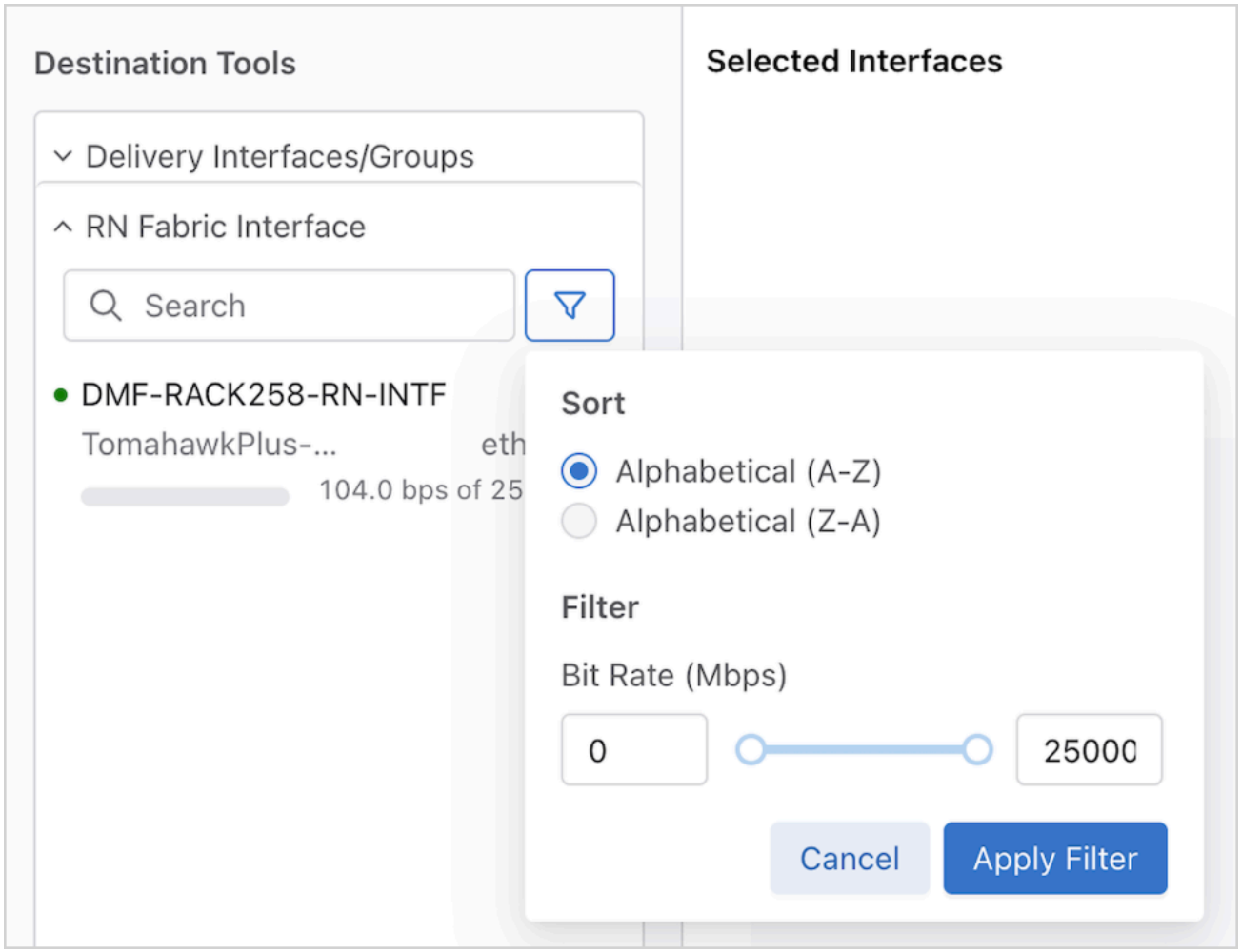
Sort Interface Groups alphabetically (A-Z, by default).

Figure 3-87: Sort Interface Groups



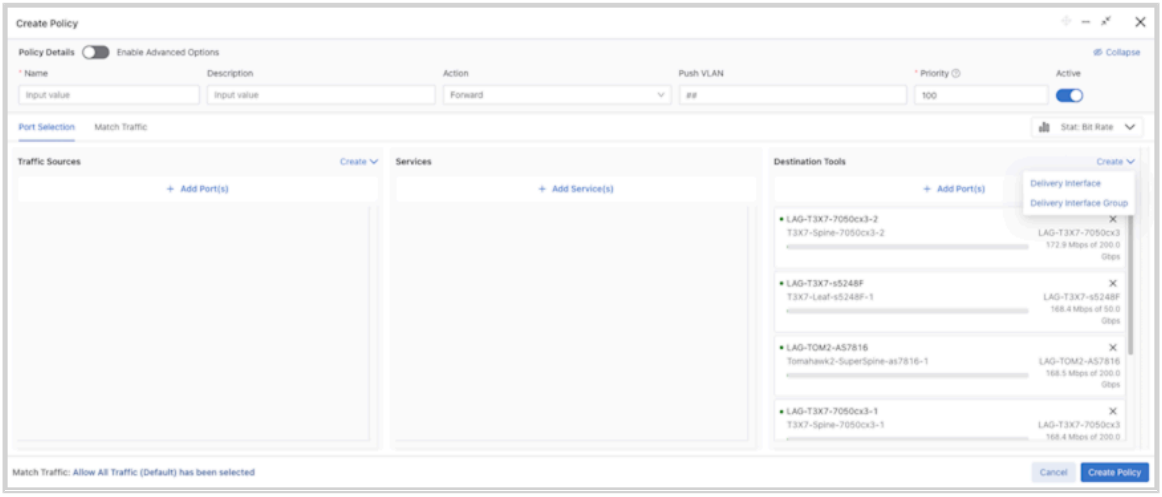
Sort Recorder Node Fabric Interfaces alphabetically (A-Z, by default) and filter by Bit Rate.

Figure 3-88: Sort Recorder Nodes



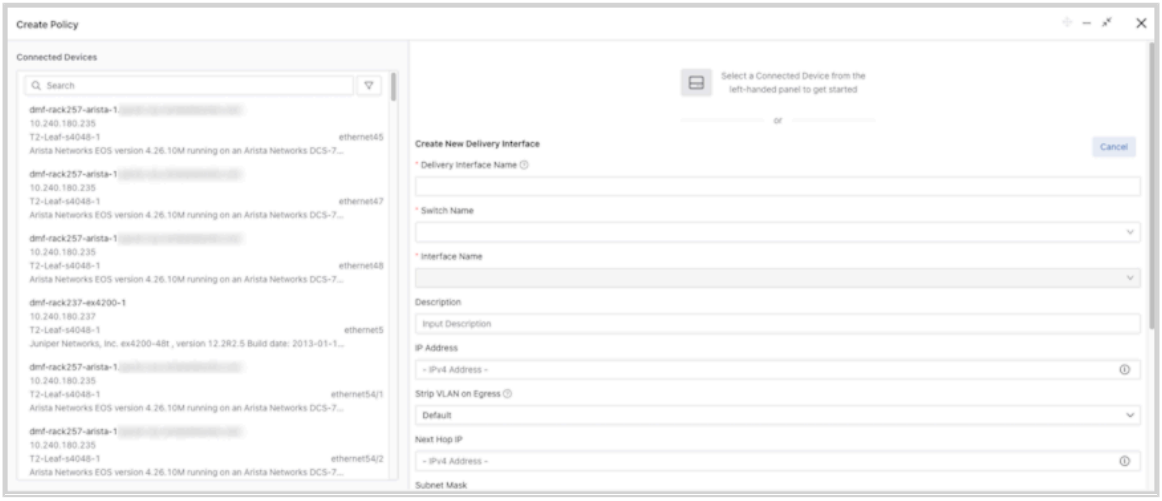
Suppose there is still a need to create **Destinations** (Delivery Interfaces). In that case, **Create** has two selections: **Create Delivery Interfaces** and **Delivery Interface Groups**.

Figure 3-89: Create Delivery Interfaces / Delivery Interface Groups



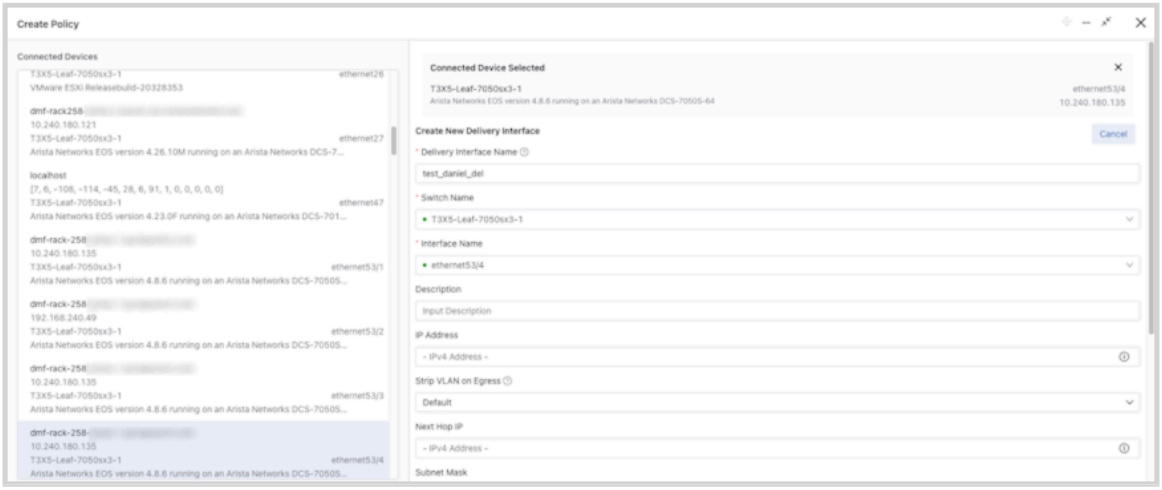
Selecting **Delivery Interface** opens a form to configure a **Delivery Interface**. Enter the required settings to configure the new **Delivery Interface**.

Figure 3-90: Configure Delivery Interface



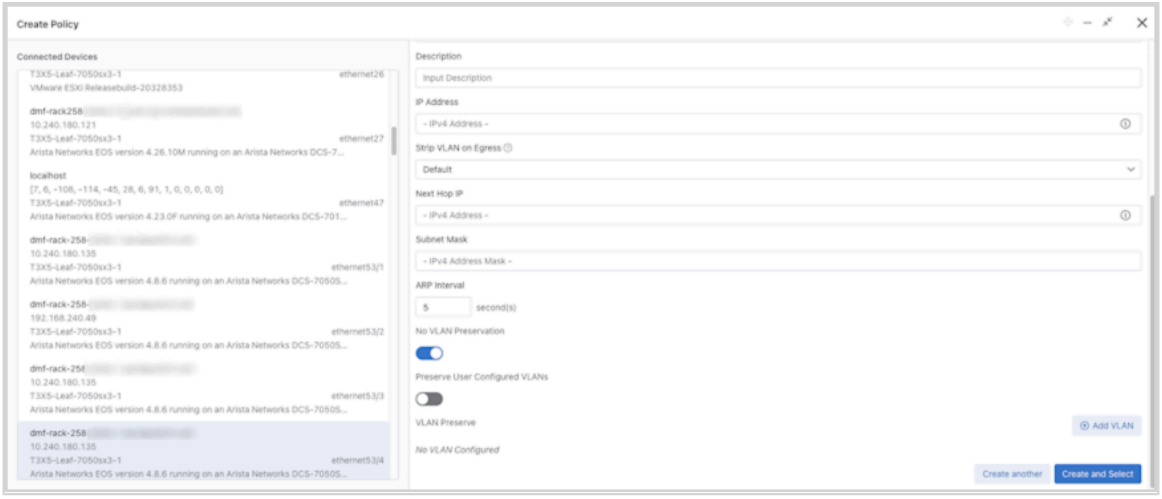
Alternatively, the left column allows the selection of an existing connected device to pre-populate the **Switch Name** and **Interface Name** fields and to configure a **Delivery Interface** based on a connected device. Select **Create and Select** to create the **Delivery Interface** and associate it with the current policy.

Figure 3-91: Associate Delivery Interface



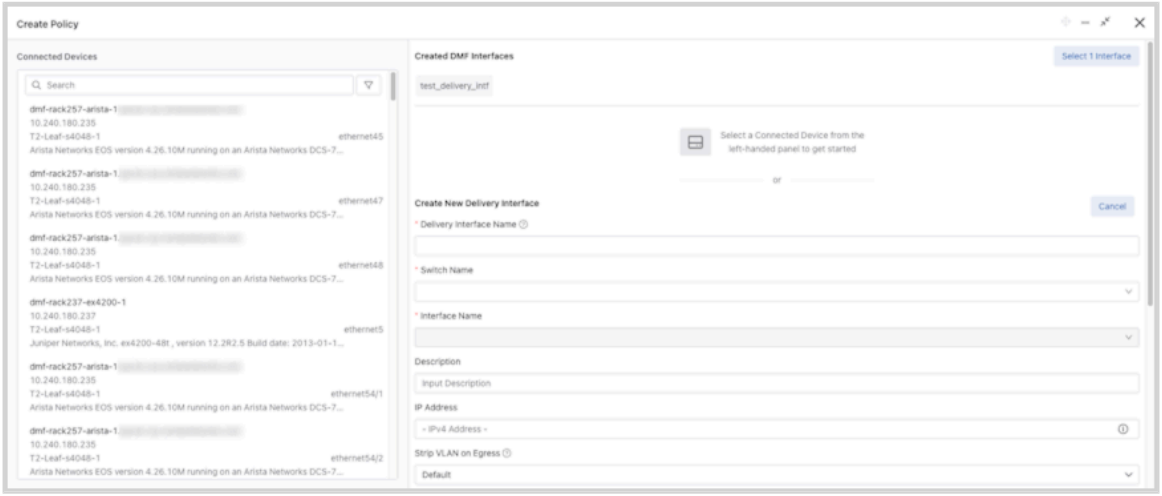
To create multiple **Delivery Interfaces**, select **Create another** to create an interface using the current configuration. This action clears the form to allow the creation of an additional **Delivery Interface**.

Figure 3-92: Multiple Delivery Interfaces



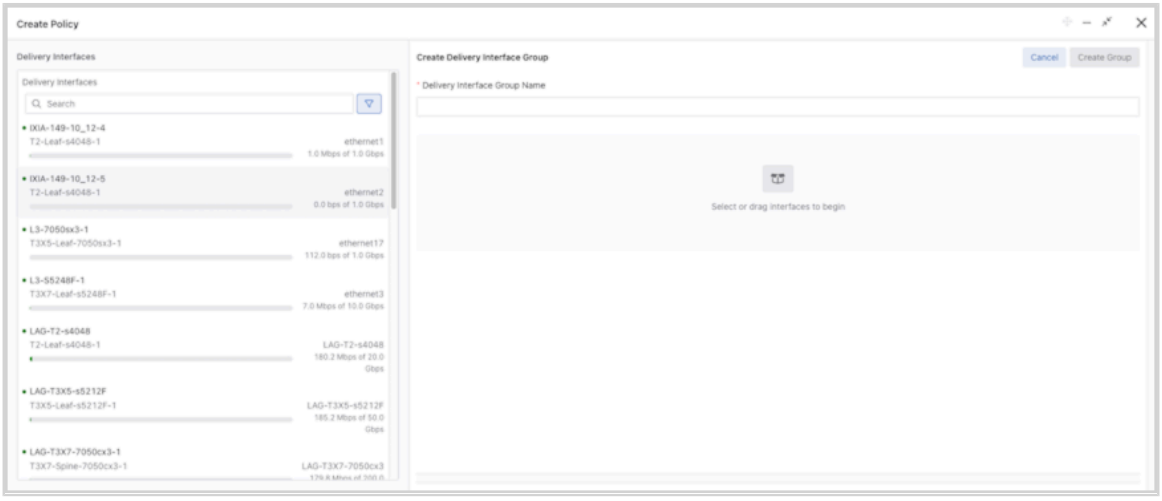
Select (n) interface associates all created **Delivery Interfaces** to the current policy.

Figure 3-93: Select Number of Interfaces & Associate



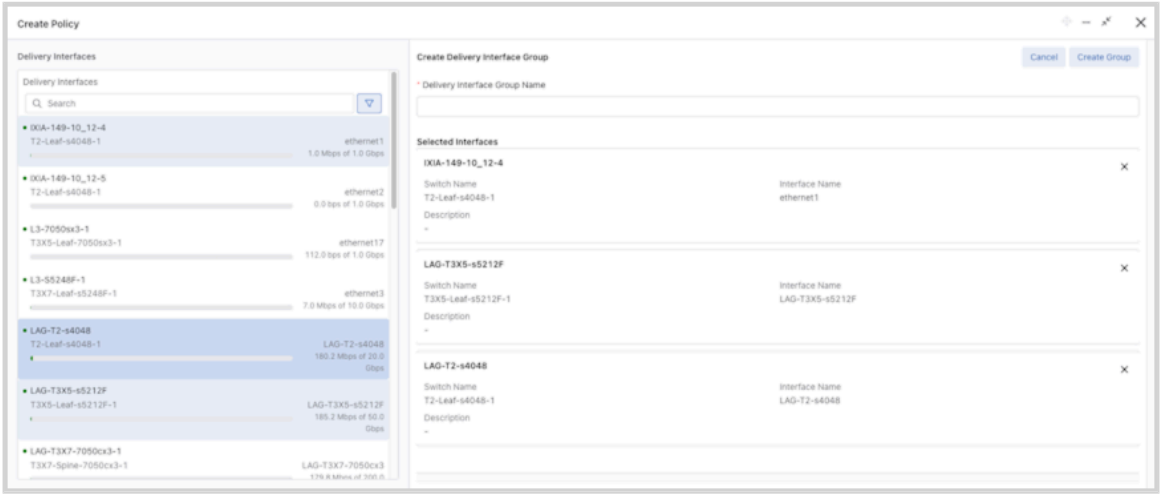
Select **Create Delivery Interface Group** to create a group of delivery interfaces.

Figure 3-94: Create Delivery Interface Group



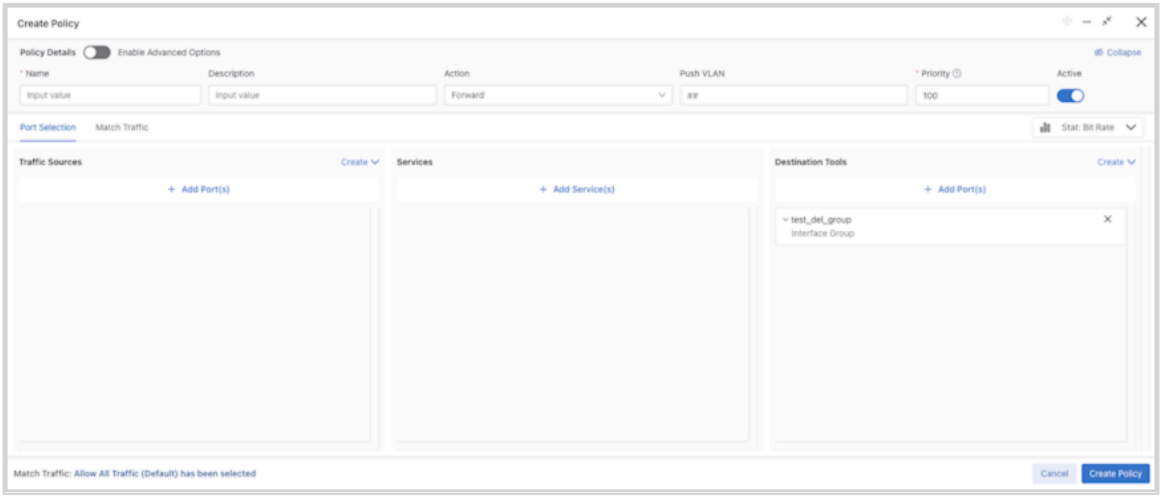
Select one or more delivery interfaces to create a **Delivery Interface Group**.

Figure 3-95: Multiple Delivery Interfaces



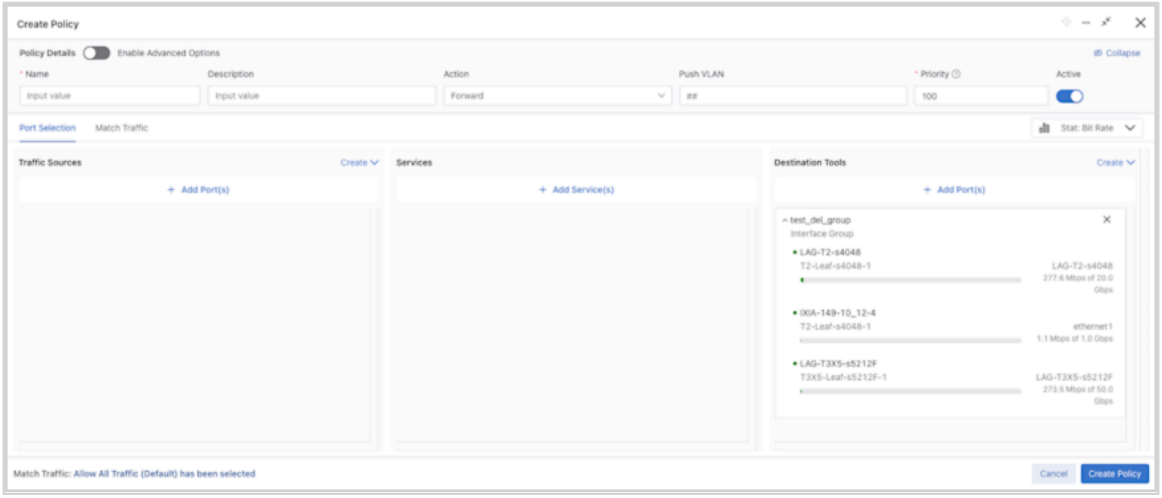
Select **Create Group** to create the **Delivery Interface Group** and associate the group with the current policy.

Figure 3-96: Associate Delivery Interface Group



Expand the group tile to view interfaces within an **Interface Group**.

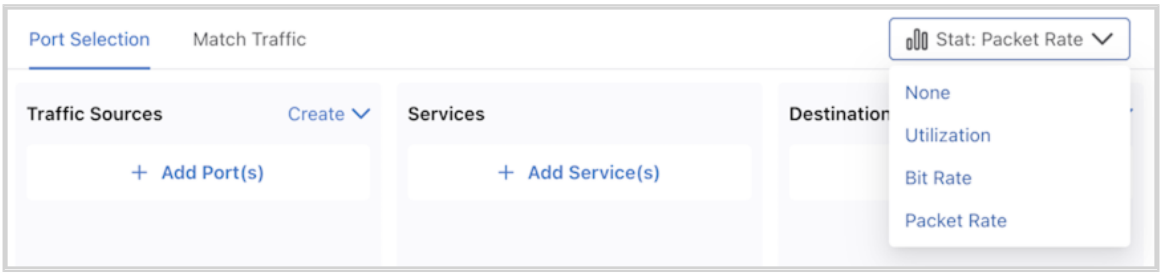
Figure 3-97: Expand Details



3.4.5 Stat Picker

Use the **Stat: Packet Rate** drop-down to select view specific data for the associated interfaces.

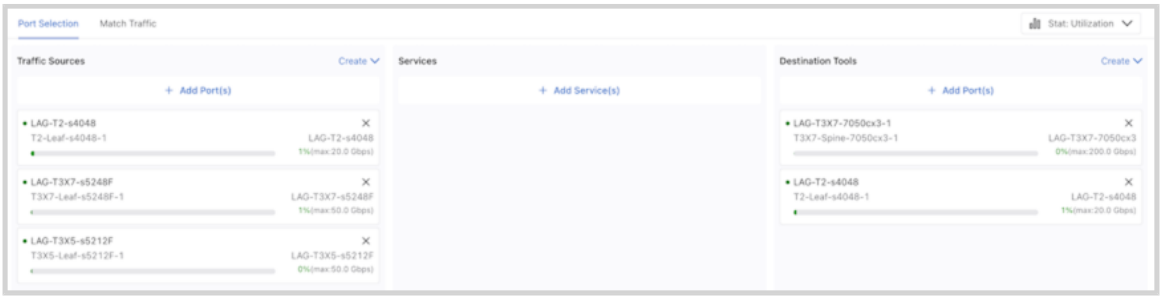
Figure 3-98: None



The data options are:

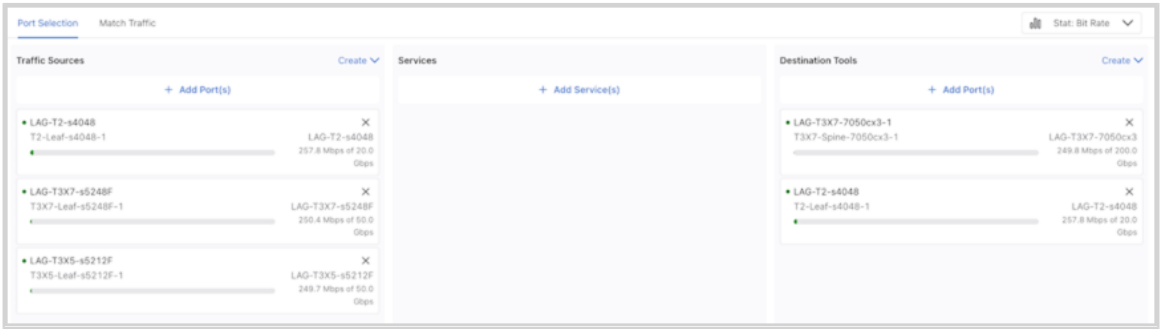
Utilization

Figure 3-99: Utilization



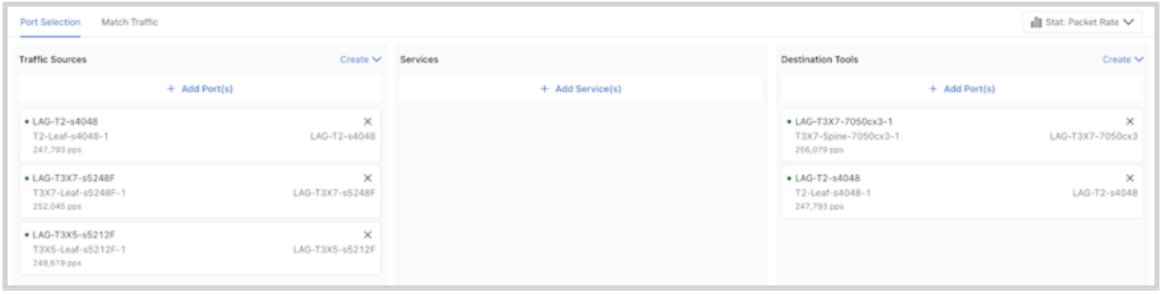
Bit Rate (default)

Figure 3-100: Bit Rate



Packet Rate

Figure 3-101: Packet Rate

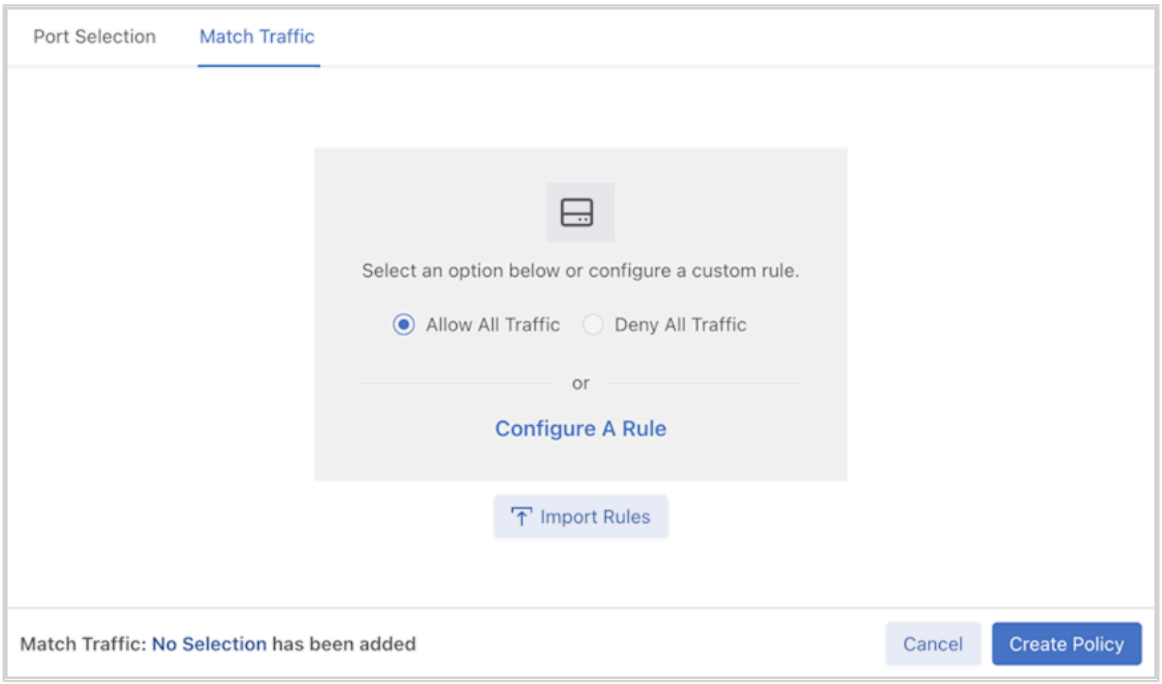


3.4.6 Match Traffic and Match Traffic Rules

Match Traffic

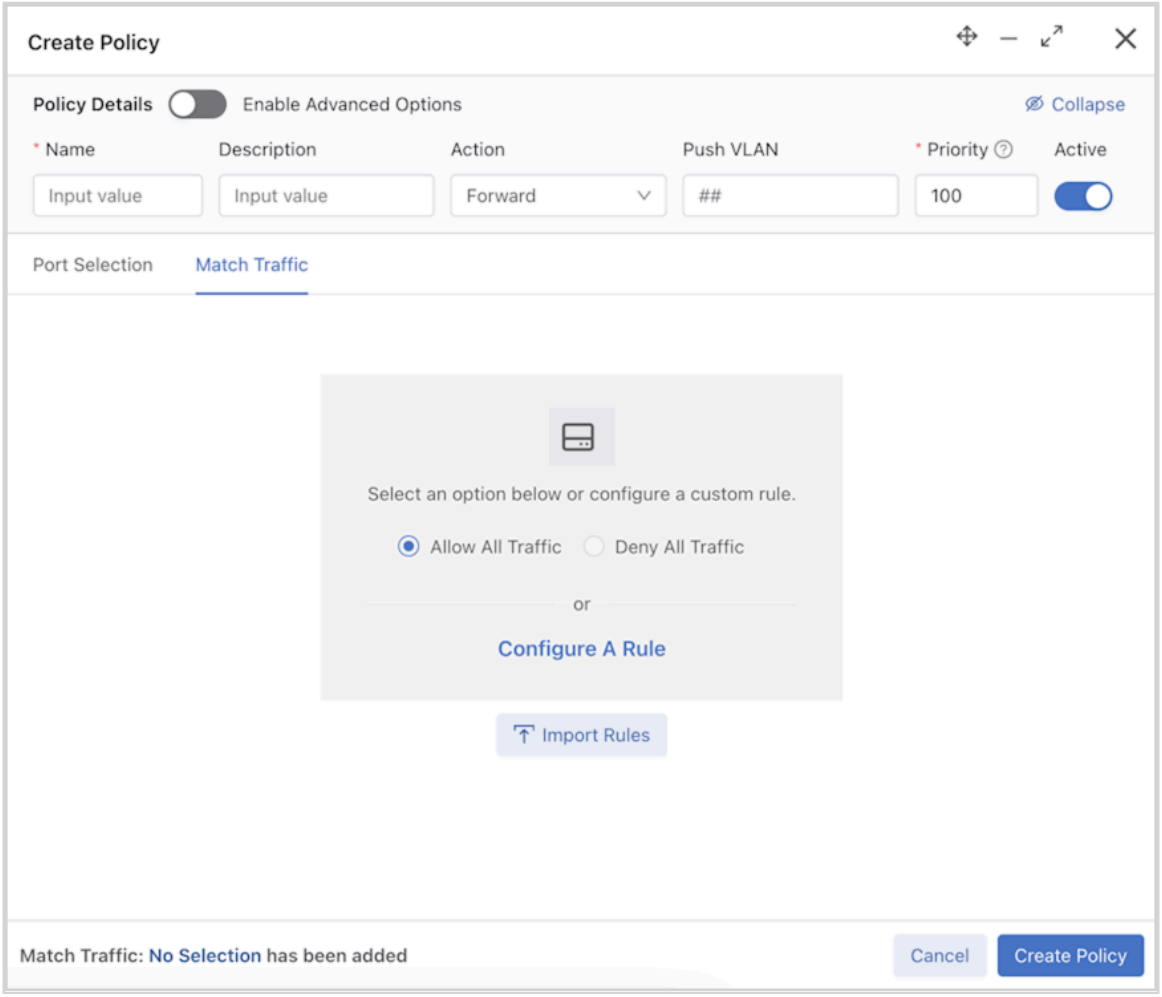
Use the **Match Traffic** tab to configure rules for the current policy.

Figure 3-102: Match Traffic



There are four options to configure traffic rules.

Figure 3-103: Configuration Options



Select the **Allow All Traffic** or **Deny All Traffic** radio button to quickly configure a rule for all traffic.

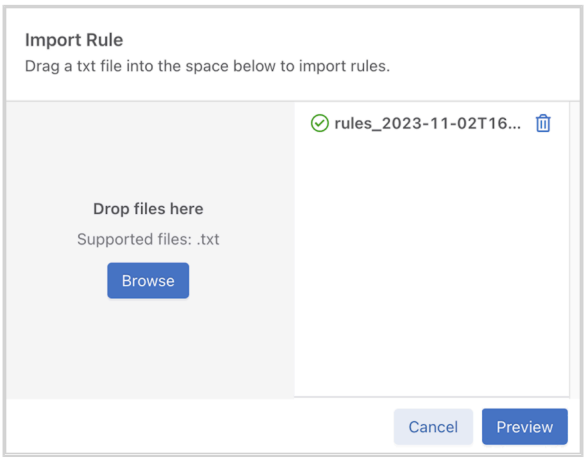
Navigate to the **Rule Details** configuration panel using **Configure A Rule**. Refer to the **Custom Rule**, **Match Rule Shortcut**, and **Match Rule Group** sections for more information.

Import Rules opens the import rule configuration dialog and supports importing `.txt` files using drag and drop or **Browse**.

Example Text File

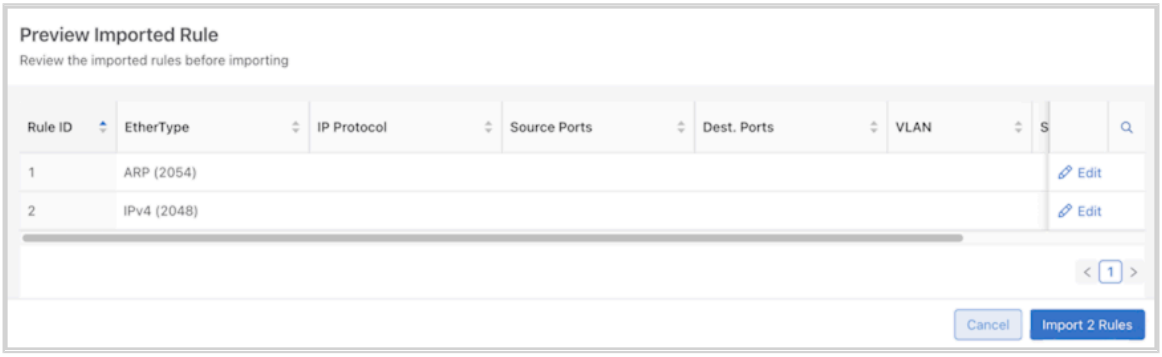
```
1 match ip
2 match tcp
3 match tcp src-port 80
4 match tcp dst-port 25
```

Figure 3-104: Import Rules



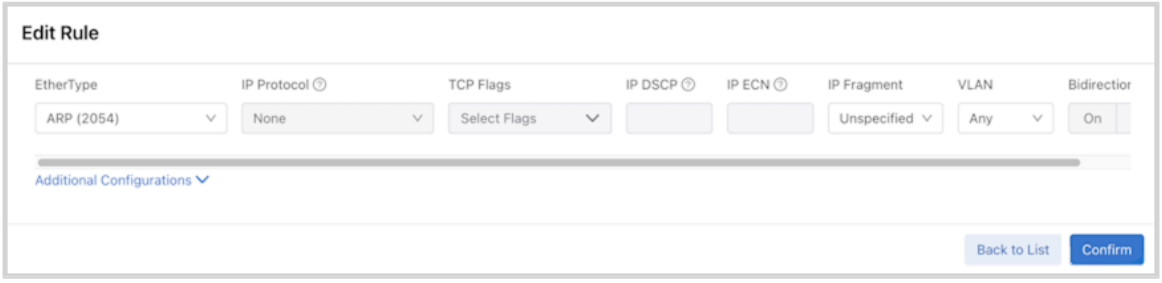
Select **Preview** to verify the import result.

Figure 3-105: Preview



While using the Preview Imported Rule table, select **Edit** to open the **Edit Rule** configuration panel.

Figure 3-106: Edit Rule

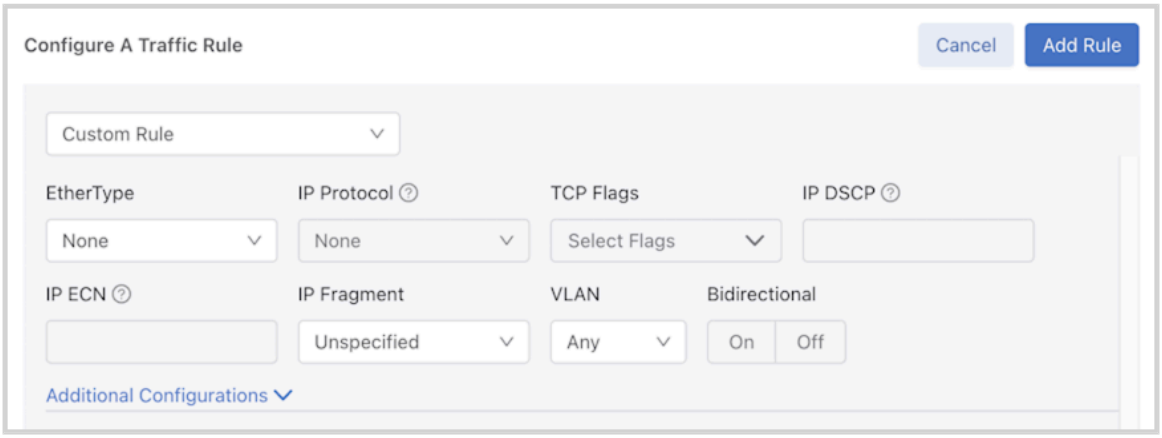


Select **Confirm** when finished, and use **Import x Rules** to import the rules.

Custom Rule

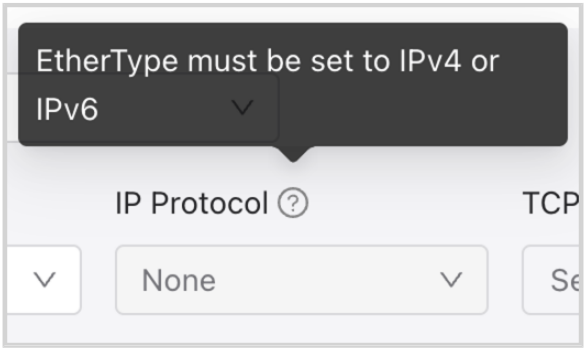
Select **Configure a Rule** to open the **Configure A Traffic Rule** window.

Figure 3-107: Configure a Traffic rule



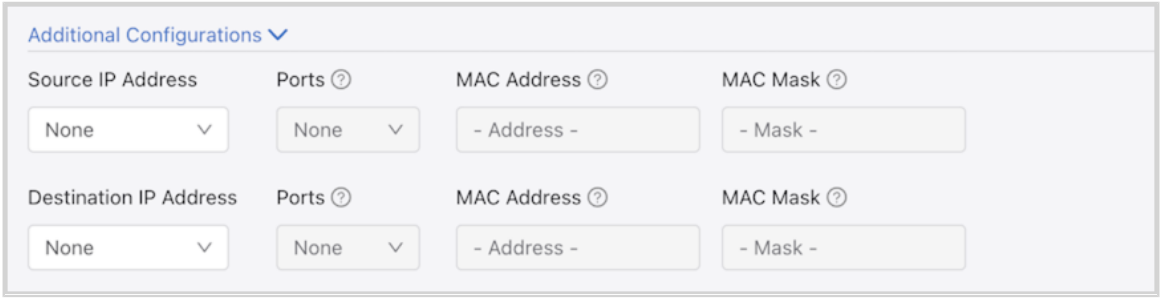
By default, the configuration method is **Custom Rule** with several fields disabled by default; hover over the **question mark** icon for more information on enabling an input field.

Figure 3-108: Help Icon



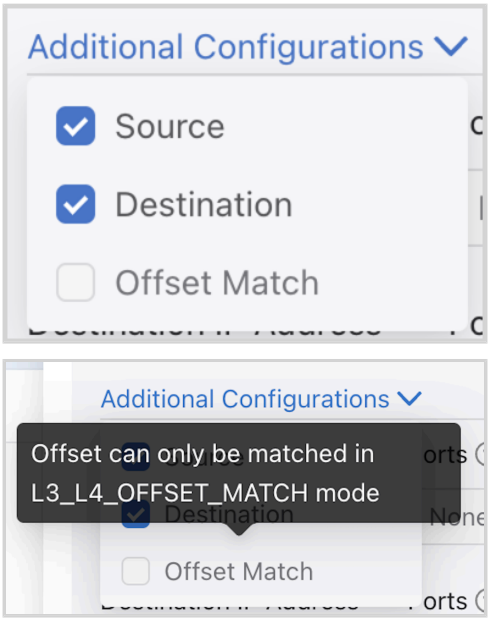
Specific **EtherTypes** will open an **Additional Configurations** panel.

Figure 3-109: Additional Configurations



Select the drop-down icon to display additional configurations (**Source**, **Destination**, **Offset Match**). Hovering over **Offset Match** allows viewing requirements to enable the **Offset Match**.

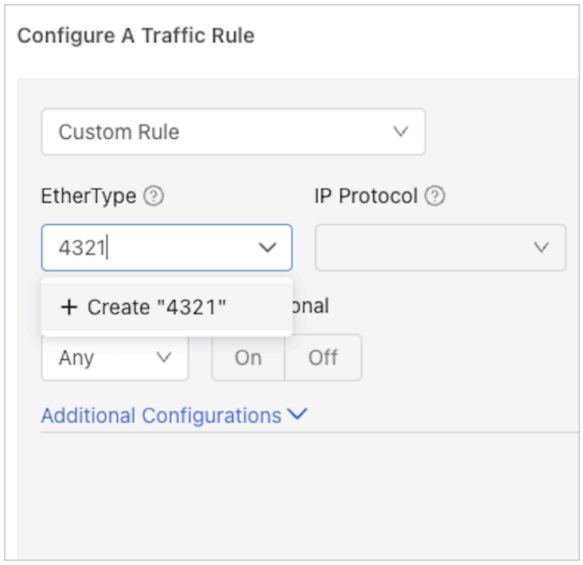
Figure 3-110: Offset Match



Custom EtherTypes

By default, the EtherType lists all known EtherType names and their hexadecimal values. DMF 8.8.0 allows custom EtherType input. The accepted input types are numbers in decimal or hexadecimal format with values greater than **1535** and less than or equal to **65535**. Begin by typing the value in the EtherType input field.

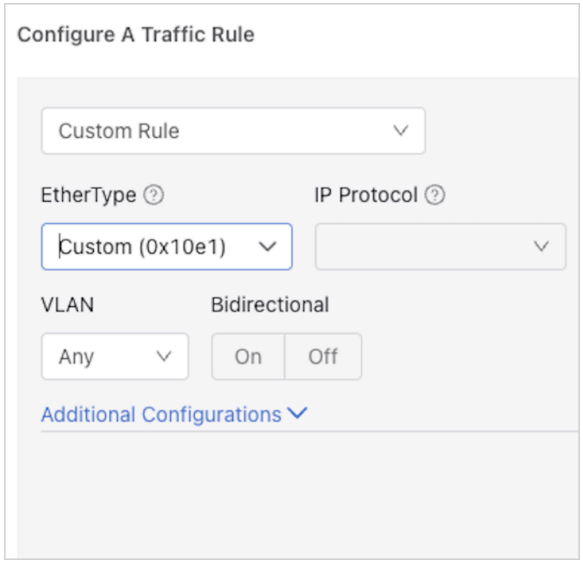
Figure 3-111: Create EtherType



Select **Create *inputted value*** to create and select the custom EtherType for the **Traffic Rule**.

Note: All values are converted to hexadecimal automatically.

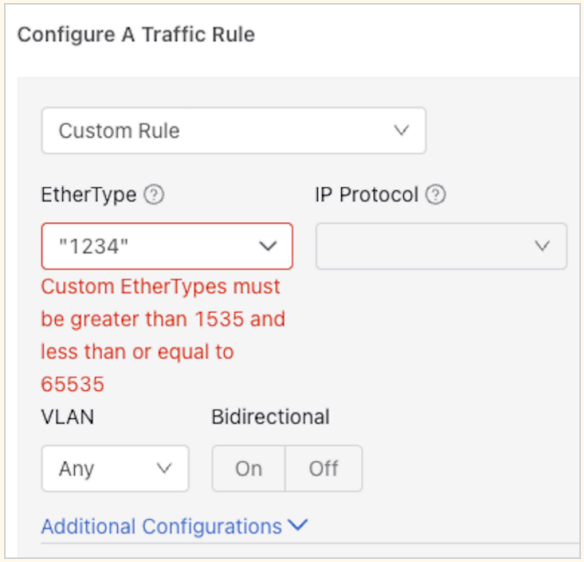
Figure 3-112: Custom Ethertype



Important:

- Only 1 EtherType can be defined for a rule.
- Creating an EtherType for a rule does not save it as a Select option for other rules. You must recreate custom rules each time they are to be used.
- EtherType must be in decimal or hexadecimal format (denoted with the prefix “**0x**”).
- EtherType must be greater than **1535** and less than or equal to **65535**. Otherwise, the system displays the following error message:

  Figure 3-113: Custom EtherType Error Message



Match Rule Shortcut

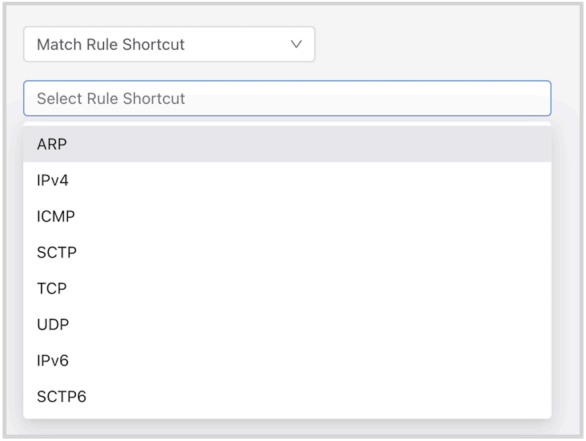
To access the **Match Rule Shortcut**, select the drop-down icon and choose **Match Rule Shortcut**.

Figure 3-114: Match Rule Shortcut



Select the **Select Rule Shortcut** selector and choose the required shortcut rules (supports multi-selection).

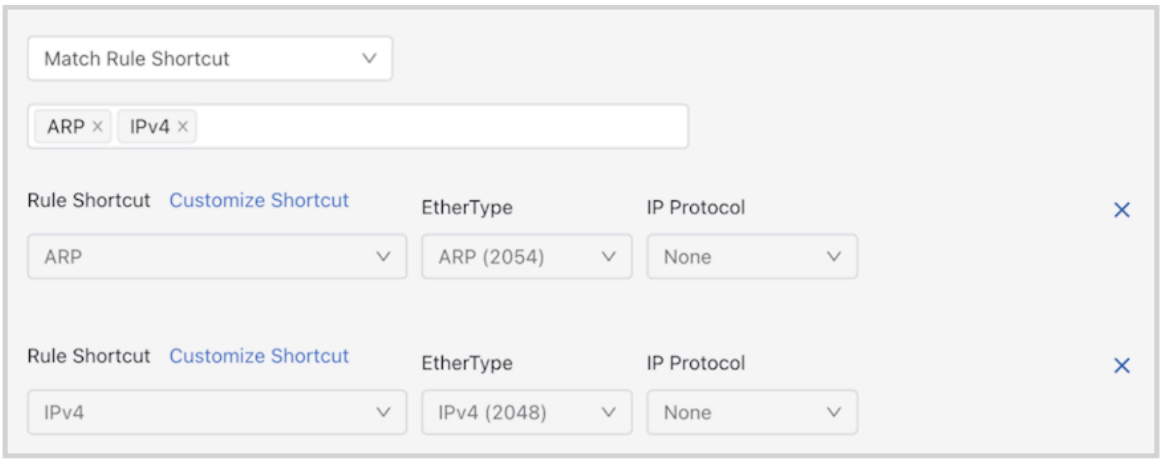
Figure 3-115: Shortcut Rule List



After selecting the rule shortcut:

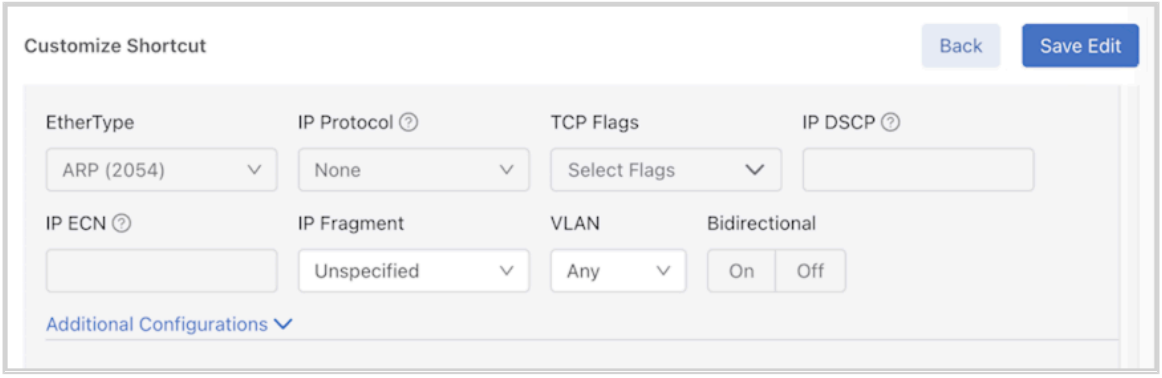
- All selected rules appear as a card in the selector.
- Delete selected rules using the **x** icons.
- Select **Customize Shortcut** to edit a rule shortcut.

Figure 3-116: Edit Shortcut



After editing, select **Save Edit** to return to the **Match Rule Shortcut** view.

Figure 3-117: Save Edits



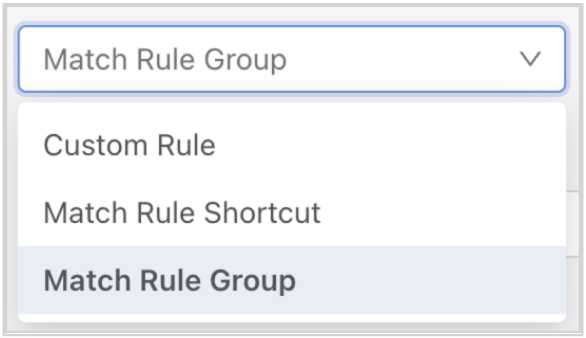
After configuring the shortcut rules, select **Add (n) Rules** to finish the configuration.



Match Rule Group

To access the **Match Rule Group**, select the drop-down and choose **Match Rule Group**.

Figure 3-118: Match Rule Group



To select a rule group, select the drop-down under **Rule Group**. All rule groups appear in the menu. Select one. There is no multi-select available. Repeat the **Match Rule Group** steps to add more than one rule group.

Figure 3-119: Rule Group List



After configuring the rule group, select **Add Rule** to finish the configuration.

Figure 3-120: Add Rule



3.4.7 Rules Table

All configured rules appear in the **Rules Table**.

Figure 3-121: Rules Table



- **Import Rules**

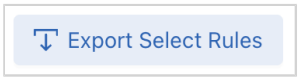
  Figure 3-122: Import Rules

  - Similar in function to **Import Rules** on the start page. Refer to Start Page -> Import Rules for more information.

- **Export Select Rules**

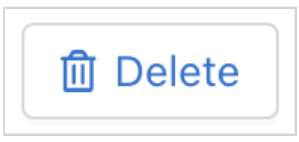
  Figure 3-123: Export Select Rules

  - Disabled by default when no rule is selected.
  - Enabled when one or more than one rule is selected.
  - Select to export selected rules information as a **`.txt`** file.

- **Delete**

  Figure 3-124: Delete

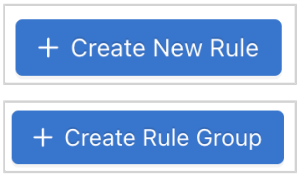
  

  - Disabled by default when no rule is selected.
  - Enabled when one or more than one rule is selected.

- Select to delete the selected rules.

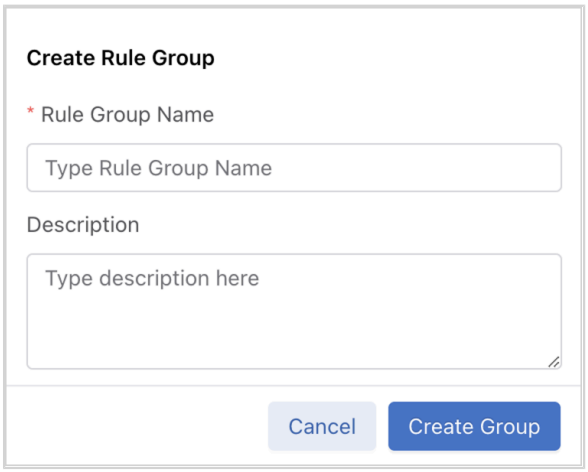
- **Create New Rule** and **Create Rule Group** buttons

  Figure 3-125: Create New Rule / Create Rule Group



  - The control appears as **Create New Rule** when no rule is selected. Select to open the **Create New Rule** screen.
  - When one or more rules are selected, the control changes to **Create Rule Group**. Select to open the **Create Rule Group** screen.

    Figure 3-126: Create Rule Group



The **Rule Group** Name is required. Select **Create Group** to confirm the rule group creation.

- **Table Actions**

  Figure 3-127: Edit / Delete



  - Select **Edit** to edit the rule view.
  - Select **Delete** to delete the rule.

- **Table Search**

  Figure 3-128: Table Search

  - The **Rules Table** supports search functionality. Select the magnifying glass icon to activate the search input fields and search the results by the context of each column.

- **Checkbox**

  Figure 3-129: Checkbox

  - Check the box to select a rule and use the function buttons described above.

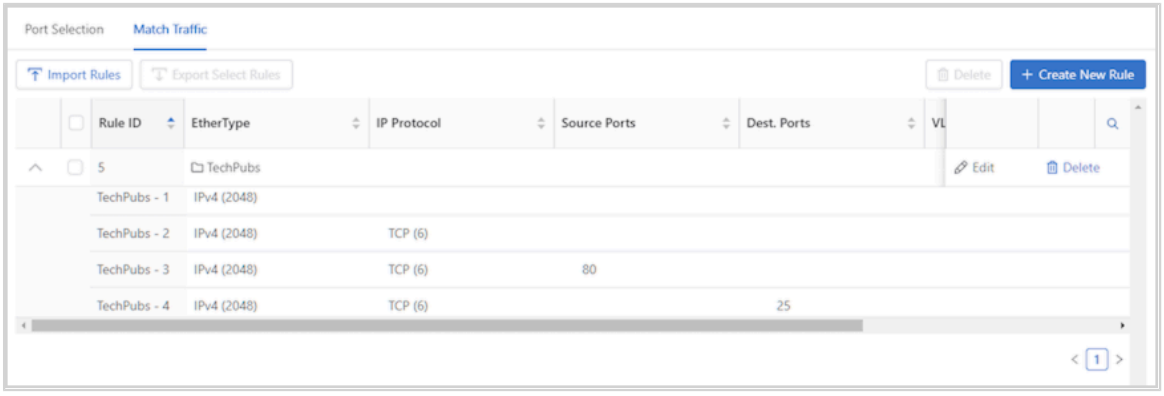
- **Expandable Group Rules**
  - **Group Rules** in the **Rule Table** display as the group's name with an expand icon.

    Figure 3-130: Expand

  - Select the **expand** icon to see the rules included in the group.

    Figure 3-131: Expanded Column

3.4.8 Persist UDF Input Format

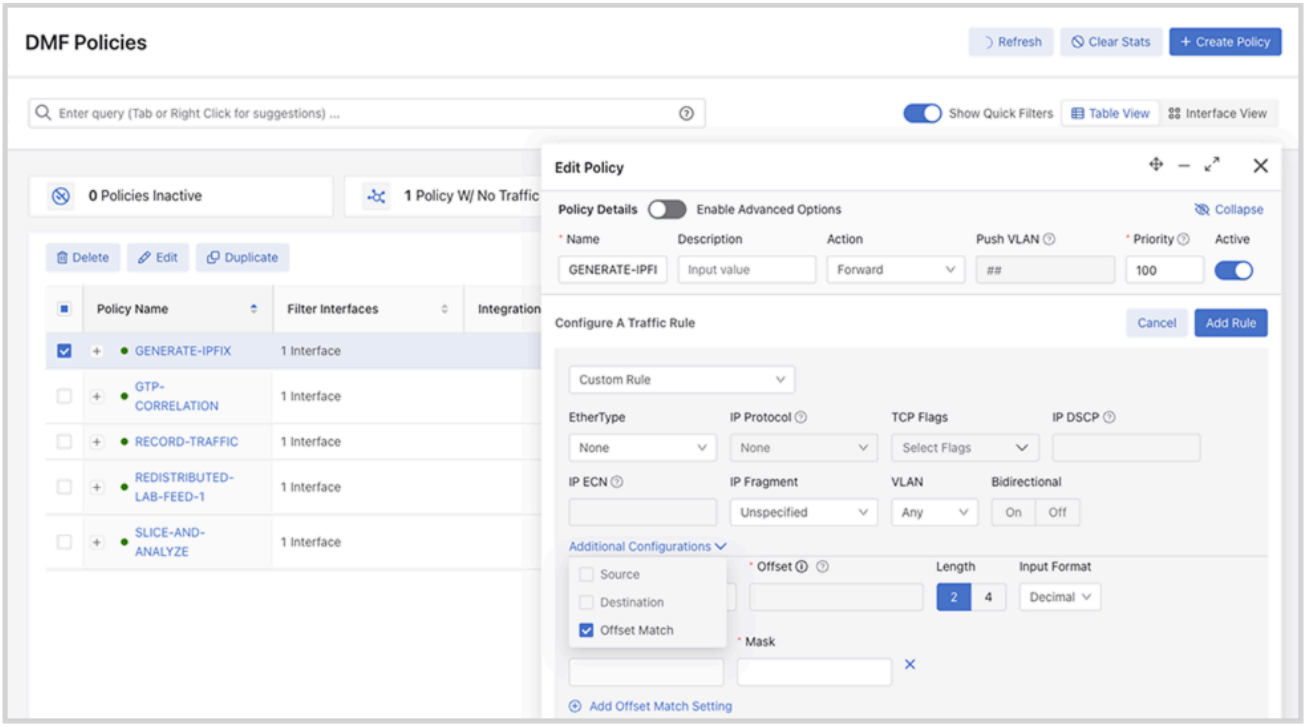
The DMF 8.7 Controller remembers the original User Defined Field (UDF) input format across clients. After configuring UDF, the input reflects the original input format, supporting cross-client consistency.

DMF supports three types of UDF input format:

- 32-bit hex string prefixed with 0x
- 32-bit unsigned integer
- Dotted Quad IPv4 address string

Navigate to **Monitoring** > **Policies**.

Figure 3-132: DMF Policies Dashboard



The workflow is contained in the **Create** or **Edit Policy** configuration under the **Match Traffic** tab.

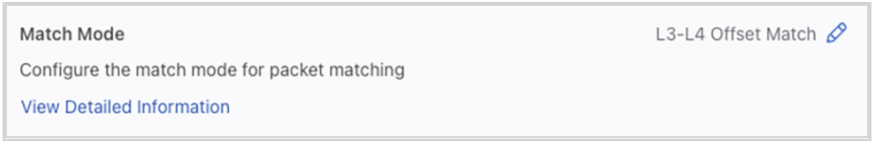
Enable **Offset Match** using the checkbox (if required). While configuring an **Offset Match** attribute in a new rule or editing an existing rule, select the **Input Format** input and enter the **Value** and **Mask** inputs in the appropriate format.

Select **Add Rule** or **Confirm Edit** to save the rule.

Use **Create Policy** or **Save Policy** to save the policy configuration.

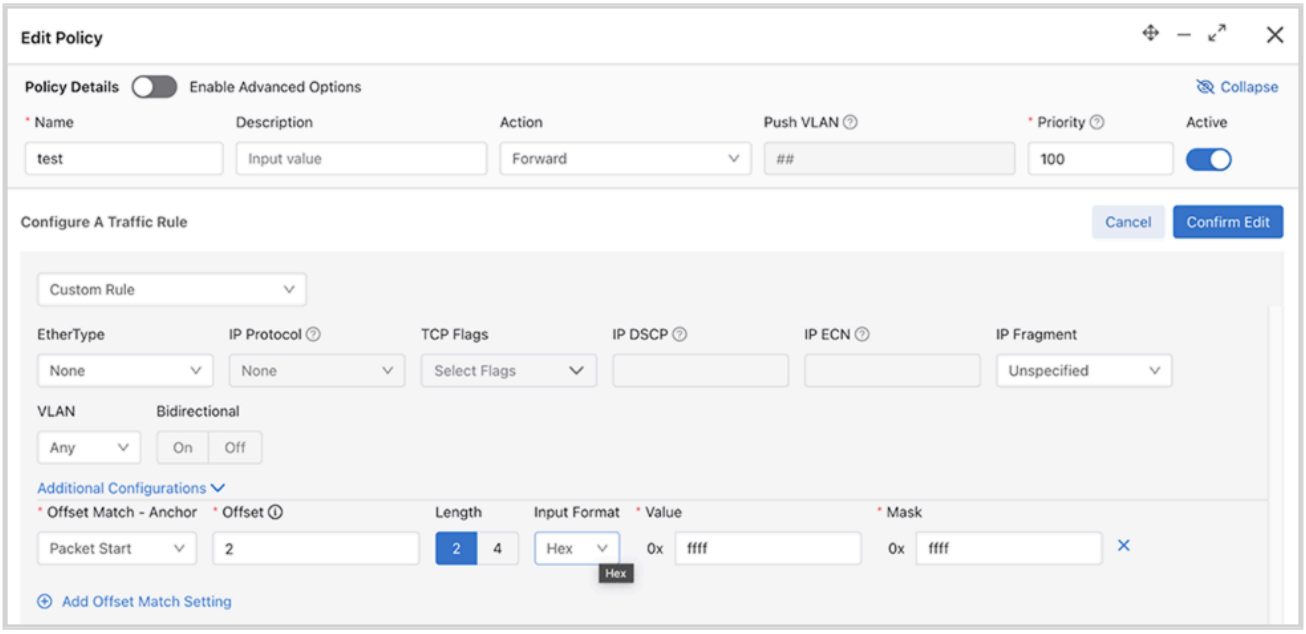
Note: The **Offset Match** is only allowed if **Match Mode** is set to **L3-L4 Offset Match** on the **DMF Features** page.

Figure 3-133: DMF Feature - Match Mode



During subsequent editing of the rule of the policy, DMF preserves and populates the **Input Format**, **Value**, and **Mask** inputs with the previously saved values.

Figure 3-134: Offset Match



Refer to the CLI show commands to view the offset matches.

3.4.8.1 Considerations

1. DMF supports only the three UDF input patterns described earlier.
2. The optimized UDFMatch contains value/mask in the user-configured format, but the value/mask of the UDFMatch may not match the configured value/mask. The Controller reports the actual internal value/mask in the form that the user configured.

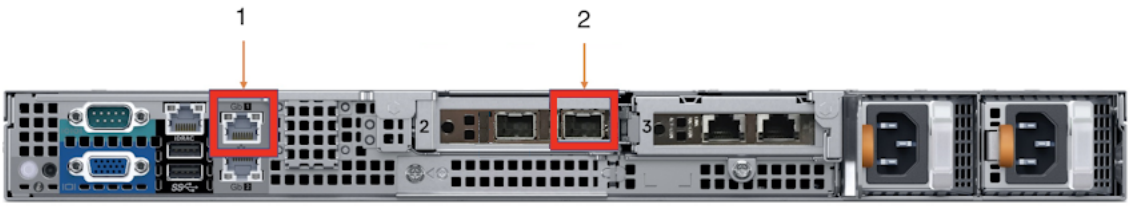
3.5 Using the Packet Capture Action in a Policy

Capture packets into a PCAP file for later processing or analysis. DANZ Monitoring Fabric (DMF) stores the captured packets on the DMF Controller hardware appliance. This feature provides a quick look at a small amount of traffic. For continuous packet capture and storage, use the DMF Recorder Node, described in the chapter DMF Recorder Node.

Note: Storing PCAP files is supported only with the hardware appliance, as running the Controller in a virtual machine is impossible. The DMF hardware appliance normally provides ***200*** GB of storage capacity, but the hardware appliance is optionally available with ***1*** TB of storage capacity.

To enable this feature, connect one of the DMF Controller hardware interfaces to a fabric switch interface defined as a DMF delivery interface.

Figure 3-135: DMF Controller Hardware Appliance (DCA-DM-CDL)

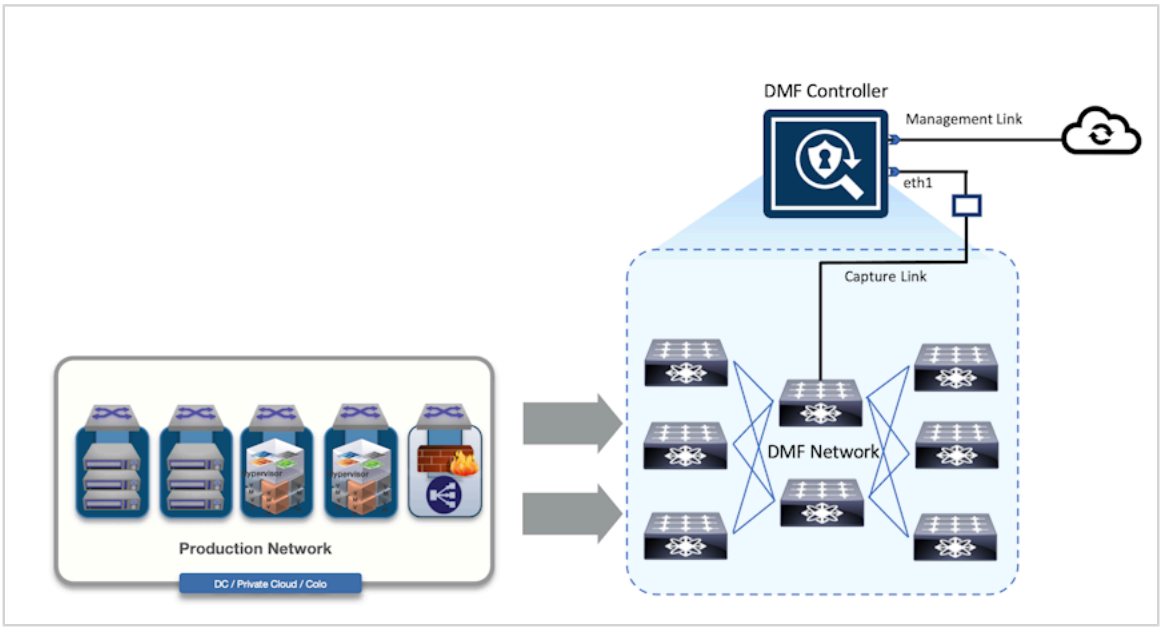


Note: The location of the DMF packet capture port varies by hardware model. The example shown in the figure is for the DCA-DM-CDL hardware appliance. Refer to the DMF Hardware Guide for the location of the DMF package capture ports.

Table 2:

1	1G Management Port	2	DMF Packet Capture

Figure 3-136: Capturing Packets on the DMF Appliance



To capture packets, define a policy with filter ports and match rules to select the interesting traffic. Specify the capture action in the policy, then schedule the policy for a duration or packet count. In the illustrated example, a service exists in the policy to modify the packets before capture, but this is optional.

By default, when the policy action is **capture**, the policy is only active after scheduling the policy. Packet captures are always saved on the master (active) Controller. In case of HA failover, previous packet captures remain on the Controller where they were initially saved.

By default, DMF automatically removes PCAP files after seven days. Change the default value using the following CLI command with the command option if preferred:

```
controller-1(config)# packet-capture retention-days <tab-key>
<retention-days> Configure packet capture file retention period in days. Default is 7 days
controller-1(config)#
```

3.6 Using the Command Line Interface

Configure a Policy

Stop, Start, and Schedule a Policy

Clear a Policy

View Policies

Rename a Policy

Define Out-of-Band Match Rules

Define a Policy with a Packet Capture Action

Rename Object

Persist UDF Input Format

3.6.1 Configure a Policy

Before configuring a policy, define the filter interfaces for use in the policy.

To configure a policy, log in to the DANZ Monitoring Fabric (DMF) console or SSH to the IP address assigned and perform the following steps:

1. From config mode, enter the **`policy`** command to name the policy and enter the config-policy submode, as in the following example:

```
controller-1(config)# policy POLICY1
controller-1(config-policy)#
```

   This example creates the policy ***POLICY1*** and enters the config-policy submode.

2. Configure one or more match rules to identify the aggregated traffic from the filter interfaces assigned to the policy, as in the following example.

```
controller-1(config-policy)# 10 match full ether-type ip dst-ip 10.0.0.50 255.255.255.255
```

   This matching rule (***10***) selects IP traffic with a destination address ***10.0.0.50***.

3. Assign one or more filter interfaces, which are monitoring fabric edge ports connected to production network TAP or SPAN ports and defined using the **`interface`** command from the ***config-switch-if*** submode.

```
controller-1(config-policy)# filter-interface TAP-PORT-1
```



Note: Define the filter interfaces used before configuring the policy.

   To include all monitoring fabric interfaces assigned the filter role, use the **all** keyword, as in the following example:

```
controller-1(config-policy)# filter-interface all
```

4. Assign one or more delivery interfaces, which monitor fabric edge ports connected to destination tools and defined using the **`interface`** command from the ***config-switch-if*** submode.

```
controller-1(config-policy)# delivery-interface TOOL-PORT-1
```

   Define the delivery interfaces used in the policy before configuring the policy. To include all monitoring fabric interfaces assigned the delivery role, use the **all** keyword, as in the following example:

```
controller-1(config-policy)# delivery-interface all
```

5. Define the action to take on matching traffic, as in the following example:

```
controller-1(config-policy)# action forward
```

   - The **forward** action activates the policy so matching traffic immediately starts being forwarded to the delivery ports identified in the policy. The other actions are **capture** and **drop**.
   - A policy is active when the configuration of the policy is complete, and a valid path exists through the network from a minimum of one of the filter ports to at least one of the delivery ports.

- When inserting a service in the policy, the policy can only become active and begin forwarding when at least one delivery port is reachable from all the post-service ports defined within the service.

To verify the operational state of the policy enter the **show policy** command.

```
controller-1# show policy GENERATE-IPFIX-NETWORK-TAP-1
Policy Name : GENERATE-IPFIX-NETWORK-TAP-1
Config Status : active - forward
Runtime Status : installed
Detailed Status : installed - installed to forward
Priority : 100
Overlap Priority : 0
# of switches with filter interfaces : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces : 0
# of filter interfaces : 1
# of delivery interfaces : 1
# of core interfaces : 0
# of services : 0
# of pre service interfaces : 0
# of post service interfaces : 0
Push VLAN : 3
Post Match Filter Traffic : -
Total Delivery Rate : -
Total Pre Service Rate : -
Total Post Service Rate : -
Overlapping Policies : none
Component Policies : none
~ Match Rules ~
# Rule
-|-----------|
1 1 match any
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF           IF Switch       IF Name    State Dir Packets   Bytes       Pkt Rate Bit Rate Counter Reset Time
-|-------------|---------------|----------|-----|---|---------|-----------|--------|--------|-------------------------------|
1 TAP-TRAFFIC-2 FILTER-SWITCH-1 ethernet16 up    rx  182876967 69995305364 0        -        2022-10-31 23:13:10.177000 PDT
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF           IF Switch       IF Name    State Dir Packets   Bytes       Pkt Rate Bit Rate Counter Reset Time
-|-------------|---------------|----------|-----|---|---------|-----------|--------|--------|-------------------------------|
1 TAP-TRAFFIC-1 FILTER-SWITCH-1 ethernet15 up    tx  182876967 69995305364 0        -        2022-10-31 23:13:10.177000 PDT
~ Service Interface(s) ~
None.
~ Core Interface(s) ~
None.
~ Failed Path(s) ~
None.
controller-1#
```

Note: If two policies have the same filter and delivery interfaces and the same priority with similar match conditions, then incorrect statistics can result for one or both policies. To alleviate this issue, either increase the priority or change the match conditions in one of the policies.

Detailed status in **show policy** command shows detailed information about a policy status. If for any reason a policy fails, the detailed status shows why the policy failed. One cause of policy failure is the TCAM reaching its total capacity. When this happens, the detailed status shows a message like **Table ing_flow2 is full <switch_DPID>**.

- ***ing_flow1***- used for programming analytics tracking like DNS, DHCP, ICMP, TCP control packets, and ARP.
- ***ing_flow2*** is the TCAM table used for programming data forwarding.
- To delete an existing policy, use the **no policy** command and identify the policy to delete, as in the following example:

```
controller-1(config-policy)# no policy policy-name-1
Warning: submode exited due to deleted object
```

- When deleting a policy, DMF deletes all traffic rules associated with the policy.

3.6.2 Stop, Start, and Schedule a Policy

Enter the **active** or **inactive** command from the **config-policy** submode to enable or disable a policy.

To stop an action that is currently active, enter the **stop** command from the **config-policy** submode for the policy, as in the following example:

```
controller-1(config)# policy policy1
controller-1(config-policy)# stop
```

By default, if the policy action is **forward** or **drop**, the policy is active unless it is manually stopped or disabled.

To start a stopped or inactive policy immediately, enter the **start now** command from the config-policy submode for the policy, as in the following example:

```
controller-1(config)# policy policy1
controller-1(config-policy)# start now
```

For a policy with the **forward** action, the **start now** command causes the policy to run indefinitely. However, policies with the **capture** action run capture for ***1*** minute unless otherwise specified, after which the policy becomes inactive. This action prevents a capture from running indefinitely and utilizes the appliance storage capacity.

Use the **start** command with other options to schedule a stopped or inactive policy. The full syntax for this command is as follows:

start { **now** [**duration** ***duration***] [**delivery-count** ***delivery-packet-count***] | **automatic** | **on-date-time** ***start-time*** [**duration** ***duration***] **seconds-from-now** ***start-time*** [**duration** ***duration***] [**delivery-count** ***delivery-packet-count***]

The following summarizes the usage of each keyword:

- **now**: start the action immediately.
- **delivery-count**: runs until the specified number of packets are delivered to all delivery interfaces.
- **seconds**: start the action after waiting for the specified number of seconds. For example, ***300***+ start the action in ***5*** minutes.
- **date-time**: starts the action on the specified date and time. Use the format %Y-%m-%dT%H:%M:%S.
- **duration**: DANZ Monitoring Fabric (DMF) assigns *60* seconds by default if no duration is specified. A value of *0* causes the action to run until it is stopped manually. When using the **delivery-count** keyword with the **capture** action, the maximum duration is ***900*** seconds.

For example, to start a policy with the **forward** action immediately and run for ***five*** minutes, enter the following command:

```
controller-1(config-policy)# start now duration 300
```

The following example starts the action immediately and stops after matching ***100*** packets:

```
controller-1(config-policy)# start now delivery-count 100
```

The following example starts the action after waiting ***300*** seconds:

```
controller-1(config-policy)# start 300+
```

3.6.3 Clear a Policy

To remove a specific DANZ Monitoring Fabric (DMF) policy, use the **no** keyword before the **policy** command, as in the following example:

```
controller-1(config)# no policy sample_policy
```

This command removes the policy ***sample_policy***.

To clear all policies at once, enter the following command:

```
controller-1(config)# clear-all-configured-policy
```

3.6.4 View Policies

To display the policies currently configured in the DANZ Monitoring Fabric (DMF) fabric, enter the **`show policy`** command, as in the following example:

```
controller-1# show policy
# Policy Name                    Action  Runtime           Status     Type  Pri Overlap Pri  Push VLAN Filter BW Delivery BW Post Match Fil Traf Delivery Traffic Services
-|------------------------------|-------|-----------------|----------|-----|-----------|----|---------|---------|---------|----------|--------|--------------------------|
1 DUPLICATE-NETWORK-TAPS         drop    installed to drop Configured 100   0           1    -         10Gbps    917Mbps                -        -
2 GENERATE-IPFIX-NETWORK-TAP-1   forward installed         Configured 100   0           3              10Gbps    10Gbps                 -        -
3 GENERATE-NETFLOW-NETWORK-TAP-1 forward installed         Configured 100   0           2              10Gbps    10Gbps                 -        -
4 GENERATE-TCP-ANALYTICS-1       forward installed         Configured 100   0           4              10Gbps    20Gbps                 -        -
5 RECORD-TCP-TRAFFIC             forward installed         Configured 100   0           5              10Gbps    10Gbps                 -        DEDUPLICATE-BEFORE-RECORD
controller-1#
```

This output provides the following information about each policy.

- #: a numeric identifier assigned to the policy.
- **Policy Name**: name of the policy.
- **Action**: Forward, Capture, or Drop.
- **Runtime Status**: a policy is active only when the policy configuration is complete, and a valid path exists through the network from a minimum of one of the filter ports to at least one of the delivery ports (and moves on through the service ports if that is specified). When inserting a service in the policy, the policy can only become active/forwarding when a delivery port is reachable from all the post-service ports of the service.
- **Type**: configured or dynamic. Refer to the Configuring Overlapping Policies section for details about dynamic policies created automatically to support overlapping policies.
- **Priority**: determines which policy is applied first.
- **Overlap Priority**: the priority assigned to the dynamic policy applied when policies overlap.
- **Push VLAN**: a feature that rewrites the outer VLAN tag for a matching packet.
- **Filter BW**: bandwidth used.
- **Delivery BW**: bandwidth used.

The following is the full command syntax for the **`show policy`** command:

`show policy` [***name*** [**filter-interfaces** | **delivery-interfaces** | **services** | **core** | **optimized-match** | **failed-paths** | **drops** | **match-rules** | **optimized-match**]]

Use the event history to determine the last time when policy flows were installed or removed. A value of **dynamic** for Type indicates the policy was dynamically created for overlapping policies.

3.6.5 Rename a Policy

Policy Renaming Procedure

> **Note:** A DANZ Monitoring Fabric (DMF) policy must exist to use the renaming feature.

Use the following procedure to rename an existing policy.

1. Use the CLI command **`policy`** ***existing-policy-name*** to enter the submode of an existing policy and then enter the **`show this`** command.

```
dmf-controller-1(config)# policy existing-policy-name
dmf-controller-1(config-policy)# show this
! policy
policy existing-policy-name
```

2. Enter the **`rename`** command with the new policy name, as shown in the following example.

```
dmf-controller-1(config-policy)# rename new-policy-name
```

> **Note:** Possible traffic loss may occur when renaming a policy.

3. Verify the policy name change using the **`show this`** command.

```
dmf-controller-1(config-policy)# show this
! policy
policy new-policy-name
dmf-controller-1(config-policy)#
```

> **Note:** A user must have permission to update the policy. The new policy name must follow the requirements for a policy name.

3.6.6 Define Out-of-band Match Rules

A policy can contain multiple match rules, each assigned a rule number. However, the rule number does not specify a priority or the sequence in applying the match rule to traffic entering the filter ports included in a policy. Instead, if the traffic matches any match rules, all actions specified in the policy are applied to all matching traffic.

The following example adds two match rules to ***dmf-policy-1***.

```
controller-1(config)# policy dmf-policy-1
controller-1(config-policy)# 10 match full ether-type ip dst-ip 10.0.0.50 255.255.255.255
controller-1(config-policy)# 20 match udp src-ip 10.0.1.1 255.255.255.0
controller-1(config-policy)# filter-interface filname2
controller-1(config-policy)# delivery-interface delname3
controller-1(config-policy)# action forward
```



Note: When changing an existing installed policy by adding or removing match rules, DANZ Monitoring Fabric (DMF) calculates the change in policy flows and only sends the difference to the switches in the path for that policy. The unmodified flows for that policy are not affected.

When more than one action applies to the same packet, DMF makes copies of the matched packet. For details, refer to the chapter Advanced Policy Configuration.

3.6.7 Define a Policy with a Packet Capture Action

Use the **`packet-capture retention-days`** command to change the number of days to retain PCAP files. To view the current setting, use the **`show packet-capture retention-days`** ***retention-days*** command.

To remove PCAP files immediately, use the **`delete packet-capture files`** command. Delete the files affiliated with a specific policy, as shown in the following example:

```
controller-1(config-policy)# delete packet-capture files policy capture file 2022-02-24-07
-31-25-34d9a85a.pcapng
```

The following command assigns the **capture** action to the current policy and schedules the packet capture to start immediately and run for ***60*** seconds.

```
controller-1(config-policy)# action capture
controller-1(config-policy)# start now duration 60
```

For a policy with the **forward** action, the **`start now`** command causes the policy to run indefinitely. However, policies with the **capture** action run capture for ***1*** minute unless otherwise specified, after which the policy becomes inactive. This action prevents a capture from running indefinitely and utilizes the appliance storage capacity.

The following command starts the capture immediately and runs until it captures ***1000*** packets:

```
controller-1(config-policy)# start now delivery-count 1000
```

Once the packet capture is complete, the PCAP file can be downloaded via HTTP using the URL displayed when entering the **`show packet-capture files`** command, as shown in the following example.

```
controller-1(config-policy)# show packet-capture files
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ All Packet Capture Files ~~~~~~~~~~~~~~~~~~~
```

To view the number of days PCAP files are retained before deletion, use the **`show packet-capture retention-days`** command as in the following example:

```
controller-1> show packet-capture retention-days
```

To view the history of packet captures, enter the following command:

```
controller-1(config-policy)# show policy capture history
# Time                          Event                          Detail             PCAP File
-|-----------------------------|------------------------------|-----------------|-----------------------------------------------|
1 2022-11-01 03:03:19.382000 PDT installation complete          capturing packets  /pcap/capture/2022-11-01-03-03-19-c106e6c.pcapng
2 2022-11-01 03:04:16.895000 PDT Configuration updated by admin. capturing packets  inactive - outside configured runtime/duration,
                                                                                    scheduled to be started in 7sec if set active
3 2022-11-01 03:04:17.266000 PDT policy removed                                     inactive - outside configured runtime/duration,
                                                                                    scheduled to be started in 6sec if set active
controller-1(config-policy)#
```

3.6.8 Rename Object

This feature provides a method to rename a DANX Monitoring Fabric (DMF) object. A DMF object must exist to use the rename feature. DMF 8.7 Controllers support the Policy rename feature.

Perform the following rename workflow steps:

1. Use the CLI to enter the submode of an existing object using the following commands:

```
dmf-controller> enable
dmf-controller# config
dmf-controller(config)#
```

2. Use the **policy** command and enter the policy name to enter the **config-policy** submode:

```
dmf-controller(config)# policy existing-policy-name
```

3. Use the **show this** command to view the details.

```
dmf-controller(config-policy)# show this

! policy
policy existing-policy-name
```

4. Use the **rename** command with the name of the new object.
   The following example illustrates changing a policy name:

```
dmf-controller(config-policy)# rename new-policy-name
dmf-controller(config-policy)# show this

! policy
policy new-policy-name
dmf-controller(config-policy)#
```

DMF updates references to the object (policy in earlier example).

If only the object name is to be updated, leaving any references to the object unmodified, use the **rename-this** command instead of **rename**, as shown in the following example:

```
dmf-controller(config)# policy existing-policy-name
dmf-controller(config-policy)# show this

! policy
policy existing-policy-name
dmf-controller(config-policy)# rename-this new-policy-name
dmf-controller(config-policy)# show this

! policy
policy new-policy-name
dmf-controller(config-policy)#
```

3.6.8.1 Limitations

- Users must have permission to update the object.
- The object must be associated with a submode (the rename command is only available within submodes).
- The new name must follow the requirements for the object's name.
- The REST API does not have a rename verb, and no REST API function can be issued to rename a policy.
- Policy rename is implemented as follows:
  1. Create a new transaction.
  2. Delete the existing object within the transaction.
  3. Create a new object with the requested name within the transaction.
  4. Commit the transaction.
- Since there is no direct support for object renaming, the Controller identifies this as a new object creation and an existing object deletion. This can result in a service interruption for semantics associated with the modified object.

3.6.9 Persist UDF Input Format

Use the CLI **show** commands to view the three offset match types.

```
dmf-controller-1># show policy p1 optimized-match
Optimized Matches :
1 ether-type 2048 offsetMatches [UdfMatch{location=UdfOffset
[anchor=l3-start, offset=4, length=2], value='0x1234', mask='0x123'}]
```

```
dmf-controller-1># show policy p1 optimized-match
Optimized Matches :
1 ether-type 2048 offsetMatches [UdfMatch{location=UdfOffset [anchor=l3-start, offset=4,
 length=2],
value='0.0.1.12', mask='0.0.2.3'}, UdfMatch{location=UdfOffset [anchor=l3-start, offset=2,
 length=2],
value='0.0.1.1', mask='0.0.12.1'}]
```

```
dmf-controller-1># show policy p1 optimized-match
Optimized Matches :
1 ether-type 2048 offsetMatches [UdfMatch{location=UdfOffset [anchor=l3-start, offset=4,
 length=2],
value='1', mask='1'}]
```

Chapter 4

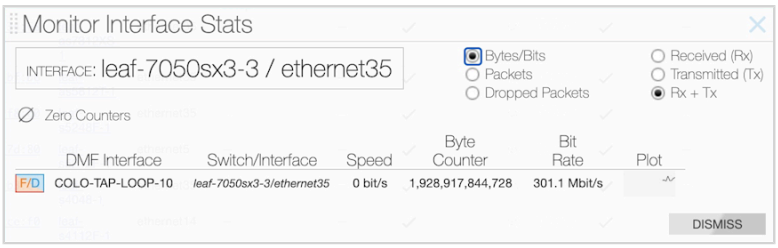
Monitoring Fabric and Production Networks

This chapter describes viewing information about the DANZ Monitoring Fabric (DMF) and connected production networks.

4.1 Monitor DMF Interfaces

Select the **Menu** control to view statistics for the specific interface and select **Monitor Stats**. The system displays the following dialog box.

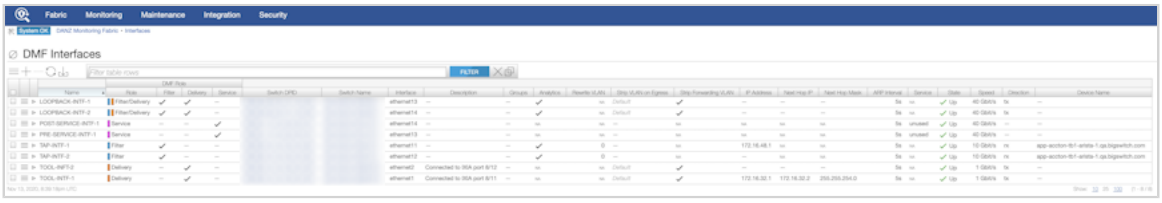
Figure 4-1: Monitor Interface Stats



This window displays statistics for up to four selected interfaces and provides a line graph (sparkline) that indicates changes in packet rate or bandwidth utilization. The auto-refresh rate for these statistics is ten seconds—mouse over the sparkline to view the range of values represented. To clear statistics for an interface, select the **Menu** control and select **Clear Stats**.

To view statistics for multiple interfaces, enable the checkbox to the left of the **Menu** control for each interface, select the **Menu** control in table, and select **Monitor Selected Stats**.

Figure 4-2: Monitoring > Interfaces



To view the interfaces assigned a specific role, use the **Monitoring > Interfaces** command and select the **Filter**, **Delivery**, or **Service** sub-option from the menu.

4.2 Viewing Oversubscription Statistics

To view peak bit rate statistics used to monitor bandwidth utilization due to oversubscription, select the **Menu** in the **Interfaces** table, select **Show/Hide Columns**, and enable the **Peak Bit Rate** checkbox on the dialog box that appears.

After enabling the **Peak Bit Rate** column, a column appears in the **Interfaces** table that indicates the relative bandwidth utilization of each interface. When using less than ***50%*** of the bandwidth, the bar appears in green; ***50-75%*** changes the bar to yellow, and over ***75%*** switches the bar color to red.

To display statistics for a specific interface, select **Monitor Stats** from the **Menu** control to the left of the row.

To reset the statistics counters, select **Clear Stats** from the **Menu** control.

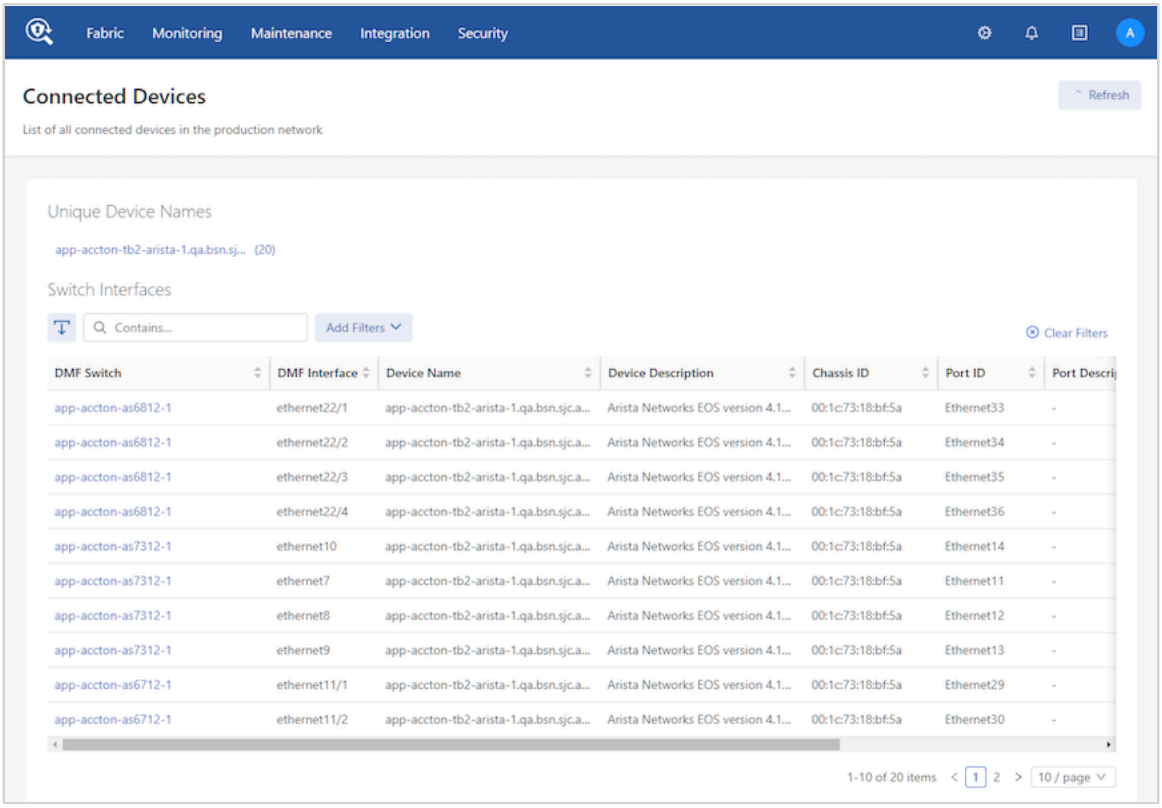


Note: DANZ Monitoring Fabric (DMF) Controllers generate SNMP traps for link saturation and packet loss. For more information, please refer to the DMF Deployment Guide - SNMP Trap Generation for Packet Drops and Link Saturation chapter.

4.3 View Fabric-Connected Devices

To view a display of the devices connected to the Controller, select **Fabric > Connected Devices** from the main menu. The system displays the following screen.

Figure 4-3: Connected Devices



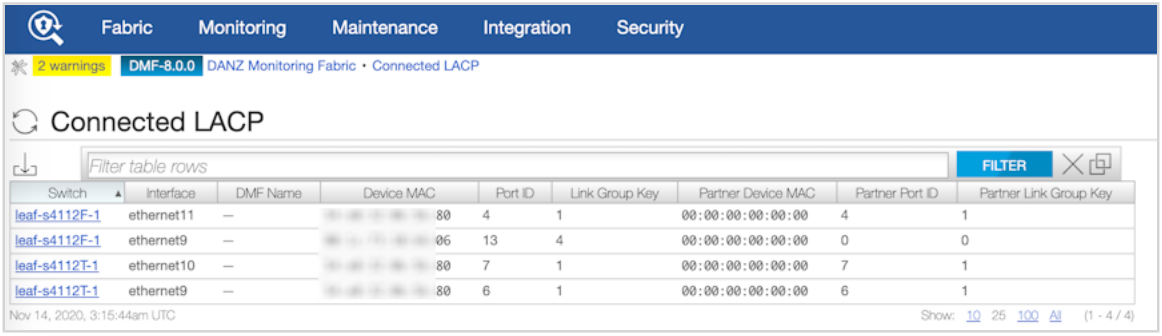
The **Switch Interfaces** table displays the unique devices connected to each out-of-band filter or delivery switch. It lists each interface's MAC address (Chassis ID) on every device connected to the fabric as a separate device.

The **Unique Device Names** table lists all unique device names with a count of interfaces in parentheses. Selecting a link in a row in this list filters the contents of the **Switch Interfaces** table.

To view a display of the devices discovered by the Controller through the Link Aggregation Control Protocol (LACP), select **Fabric > Connected LACP** from the main menu.

The system displays the following screen.

Figure 4-4: Connected LACP



This page displays the devices discovered by the Controller through LACP.

4.4 Using the Command Line Interface

Monitor Interface Configuration

View Switch Configuration

View Connected Devices and LAGs

Viewing Information about a Connected Production Network

4.4.1 Monitor Interface Configuration

To display the currently configured interfaces, enter the **`show interface-names`** command, as shown in the following example.

```
Ctrl-2> show interface-names

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~
# DMF IF      Switch              IF Name   Dir State Speed  VLAN Tag Analytics Ip address
 Connected Device
-|-----------|-------------------|---------|---|-----|------|--------|---------|----------|------
----------|
1 Lab-traffic Arista-7050SX3-T3X5 ethernet7 rx  up    10Gbps 0        True

~ Delivery Interface(s) ~
None.


~ Service Interface(s) ~
None.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Recorder Fabric Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF          Switch              IF Name    Dir           State Speed  Connected Device
-|---------------|-------------------|----------|-------------|-----|------|------------------|
1 PR-NewHW-Intf   Arista-7050SX3-T3X5 ethernet25 bidirectional up    25Gbps PR-NewHW   ens1f0
2 RMA-CNrail-intf Arista-7050SX3-T3X5 ethernet35 bidirectional up    25Gbps RMA-CNrail ens1f0
```



Note: The name is used when configuring a policy.

To display a summary of the current DANZ Monitoring Fabric (DMF) configuration, enter the **show fabric** command, as in the following example.

```
controller-1# show fabric
~~~~~~~~~~~~~~~~~~~~ Aggregate Network State ~~~~~~~~~~~~~~~~~~~~
Number of switches : 3
Inport masking : False
Start time : 2018-03-16 15:42:43.322000 PDT
Number of unmanaged services : 0
Filter efficiency : 0:1
Number of switches with service interfaces : 0
Total delivery traffic (bps) : 168bps
Number of managed service instances : 2
Number of service interfaces : 0
Match mode : l3-l4-offset-match
Number of delivery interfaces : 6
Max pre-service BW (bps) : 20Gbps
Auto VLAN mode : push-per-policy
Number of switches with delivery interfaces : 2
Number of managed devices : 1
Uptime : 5 hours, 4 minutes
Total ingress traffic (bps) : 160bps
Max filter BW (bps) : 221Gbps
Auto Delivery Interface Strip VLAN : True
Number of core interfaces : 12
Overlap : True
Number of switches with filter interfaces : 2
State : Enabled
Max delivery BW (bps) : 231Gbps
Total pre-service traffic (bps) : 200bps
Track hosts : True
Number of filter interfaces : 5
Number of active policies : 2
Number of policies : 5
~~~~~~~~~~~~~ Aggregate Interface Statistics ~~~~~~~~~~~~~
# Interface Type     Dir Packets Bytes  Pkt Rate Bit Rate
-|------------------|---|-------|------|--------|--------|
1 Filter Interface   rx  2444    455611 0        160bps
2 Delivery Interface tx  4050    421227 0        168bps
--------------------example truncated--------------------
controller-1#
```

4.4.2 View Switch Configuration

To verify the switch interface configuration, enter the **`show topology`** command, as shown in the following example.

```
controller> show topology
~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s) ~~~~~~~~~~~~~~~~~~~~~
# DMF IF    Switch IF   Name     state speed   Connected Device
-|---------|-----------|--------|-----|-------|----------------|
1 f1        filter-sw-1 s11-eth1 up    10 Gbps
2 f2        filter-sw-1 s11-eth2 up    10 Gbps
~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s) ~~~~~~~~~~~~~~~~~~~~
# DMF IF    Switch IF   Name     state speed   Connected Device
-|---------|-----------|--------|-----|-------|----------------|
1 d1        filter-sw-2 s12-eth1 up    10 Gbps
2 d2        filter-sw-2 s12-eth2 up    10 Gbps
~~~~~~~~~~~~~~~~~~~~~~~~ Service Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF           Switch IF Name    Dir state speed   Connected Device
-|----------------|---------|-------|---|-----|-------|----------------|
1 post-serv-intf-1 core-sw-1 s9-eth2     up    10 Gbps
2 pre-serv-intf-1  core-sw-1 s9-eth1     up    10 Gbps
~~~~~~~~~~~~~~~~~~~~~~~ Core Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~
# Src Switch    Src IF   Src Speed Dst Switch    Dst IF   Dst Speed
-|-------------|--------|---------|-------------|--------|---------|
1 core-sw-3     s13-eth2 10 Gbps   delivery-sw-2 s15-eth3 10 Gbps
2 core-sw-3     s13-eth1 10 Gbps   delivery-sw-1 s14-eth3 10 Gbps
3 filter-sw-1   s11-eth3 10 Gbps   core-sw-2     s10-eth1 10 Gbps
4 core-sw-2     s10-eth1 10 Gbps   filter-sw-1   s11-eth3 10 Gbps
5 delivery-sw-2 s15-eth3 10 Gbps   core-sw-3     s13-eth2 10 Gbps
6 core-sw-2     s10-eth2 10 Gbps   filter-sw-2   s12-eth3 10 Gbps
7 filter-sw-2   s12-eth3 10 Gbps   core-sw-2     s10-eth2 10 Gbps
8 delivery-sw-1 s14-eth3 10 Gbps   core-sw-3     s13-eth1 10 Gbps
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Statistics ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#  DMF IF    Switch IF     Role     State   Packets         Bytes  Pkt Rate Bit Rate
--|---------|-------------|--------|--------|-----|-------|------|--------|--------|
1 f1         filter-sw-1   s11-eth1 filter   up    0       0      0        -
2 f2         filter-sw-1   s11-eth2 filter   up    0       0      0        -
3 d1         filter-sw-2   s12-eth1 delivery up    8       600    0        -
4 d2         filter-sw-2   s12-eth2 delivery up    8       600    0        32 bps
6 -          core-sw-3     s13-eth1 core     up    3432    257400 0        32 bps
7 -          delivery-sw-2 s15-eth3 core     up    3431    257325 0        32 bps
8 -          delivery-sw-1 s14-eth3 core     up    3430    257250 0        32 bps
9 -          core-sw-2     s10-eth1 core     up    3429    257175 0        32 bps
10 -         filter-sw-1   s11-eth3 core     up    3431    257325 0        32 bps
11 -         core-sw-3     s13-eth2 core     up    3432    257400 0        32 bps
12 -         filter-sw-2   s12-eth3 core     up    3429    257175 0        32 bps
```

4.4.3 View Connected Devices and LAGs

Some information on devices in the production network, discovered using LLDP and CDP, can be seen using the **show connected-devices** command. The data helps determine if filter interfaces are connected to the intended production device.

The **show connected-devices**command from login mode displays the devices connected to the DANZ Monitoring Fabric (DMF). This command displays information about devices connected to DMF switch interfaces. DMF extracts the information from link-level protocol packets such as LLDP, CDP, and UDLD and ignores expired link-level data.

```
controller-1# show connected-devices switch FILTER-SWITCH-1
# Switch          IF Name   DMF name SPAN? Device Name          Device Description                                                
```

4.4.4 Viewing Information about a Connected Production Network

Once the monitoring fabric is set up and connected to packet feeds from the production network, DANZ Monitoring Fabric (DMF) starts to gather information about the production network. By default, DMF provides a view of all hosts in the production network visible from the filter interfaces. View this information on the GUI page under **Monitoring > Host Tracker**. As shown below, the output displays the host MAC address, IP address, when and on which filter interface traffic from the host was seen, and DHCP lease information. To display this information, enter the **`show tracked-hosts`** command, as shown in the following example.

```
# show tracked-hosts
#   IP Address      MAC Address                                       Host name                       Filter interfaces        VLANs   Last seen
 Extra info
---|---------------|-------------------------------------------------|-------------------------------|------------------------|-------|---------|
1   10.0.0.3        40:a6:d9:7c:9f:9f                                 Apple                           wireless-poe-1           0       1 hours
2   10.0.0.6        98:fe:94:1c:37:06                                 Apple                           wireless-poe-1           0       42 min
3   10.0.0.6        dc:2b:61:81:64:45                                 Apple                           wireless-poe-1           0       3 hours
4   10.0.0.7        20:c9:d0:48:f3:3d                                 Apple                           wireless-poe-1           0       2 hours
5   10.0.0.11       60:03:08:9b:4f:48                                 Apple                           wireless-poe-1           0       13 min
6   10.0.1.3        14:10:9f:e4:e6:bf                                 Apple                           wireless-poe-1           0       51 min
-------------------------------------------------------------------output truncated-------------------------------------------------------------------
```

DMF also tracks the DNS names of hosts by capturing and analyzing packets using several different protocols. To manage host-name tracking, from config-analytics mode, use the **`track`** command, which has the following syntax:

[**no**] **`track`** { **arp** | **dns** | **dhcp** | **icmp** }

For example, the following command enables tracking using DNS:

```
controller-1(config)# analytics
controller-1(config-analytics)# track dns
```



Note: DNS traffic will not be included in DMF policies when enabling DNS for tracking.

Exclude host tracking for a specific filter interface using the **no-analytics** option with the **`role`** command.

```
controller-1(config)# switch DMF-FILTER-SWITCH-1
controller-1(config-switch)# interface ethernet20
controller-1(config-switch-if)# role filter interface-name TAP-PORT-01 no-analytics
```

This command disables all host tracking on interface ***TAP-PORT-01***.

Chapter 5

DMF Service Node Appliance

This chapter describes configuring the managed services provided by the DANZ Monitoring Fabric (DMF) Service Node Appliance.

5.1 Overview

The DANZ Monitoring Fabric (DMF) Service Node has multiple interfaces connected to traffic for processing and analysis. Each interface can be programmed independently to provide any supported managed-service actions.

To create a managed service, identify a switch interface connected to the service node, specify the service action, and configure the service action options.

Configure a DMF policy to use the managed service by name. This action causes the Controller to forward traffic the policy selects to the service node. The processed traffic is returned to the monitoring fabric using the same interface and sent to the tools (delivery interfaces) defined in the DMF policy.

If the traffic volume the policy selects is too much for a single service node interface, define an LAG on the switch connected to the service node, then use the LAG interface when defining the managed service. All service node interfaces connected to the LAG are configured to perform the same action. The traffic the policy selects is automatically load-balanced among the LAG member interfaces and distributes the return traffic similarly.

5.2 Changing the Service Node Default Configuration

Configuration settings are automatically downloaded to the service node from the DANZ Monitoring Fabric (DMF) Controller to eliminate the need for box-by-box configuration. However, the option exists to override the default configuration for a service node from the ***config-service-node*** submode for any service node.



Note: These options are available only from the CLI and are not included in the DMF GUI.

To change the CLI mode to **config-service-node**, enter the following command from config mode on the Active DMF controller:

```
controller-1(config)# service-node service_node_alias
controller-1(config-service-node)#
```

Replace ***service_node_alias*** with the alias to use for the service node. This alias is affiliated with the hardware MAC address of the service node using the **mac** command. The hardware MAC address configuration is mandatory for the service node to interact with the DMF Controller.

Use any of the following commands from the **config-service-node** submode to override the default configuration for the associated service node:

- **admin password**: set the password to log in to the service node as an admin user.
- **banner**: set the service node pre-login banner message.
- **description**: set a brief description.
- **logging**: enable service node logging to the Controller.
- **mac**: configure a MAC address for the service node.
- **ntp**: configure the service node to override default parameters.
- **snmp-server**: configure an SNMP trap host to receive SNMP traps from the service node.

5.3 Using SNMP to Monitor DPDK Service Node Interfaces

Directly fetch the counters and status of the service node interfaces handling traffic (DPDK interfaces). The following are the supported OIDs.

```
interfaces MIB: #.1.3.6.1.2.1.2#
ifMIBObjects MIB: #.1.3.6.1.2.1.31.1#
```



Note: A three-digit number between ***101*** and ***116*** identifies SNI DPDK (traffic) interfaces.

In the following example, interface ***sni5*** (105) handles data traffic. To fetch the packet count, use the following command:

```
snmpget -v2c -c public 10.106.6.5 .1.3.6.1.2.1.31.1.1.1.6.105
IF-MIB::ifHCInOctets.105 = Counter64: 10008
```

To fetch the counters for packets exiting the service node interface, enter the following command:

```
snmpget -v2c -c public 10.106.6.5 .1.3.6.1.2.1.31.1.1.1.10.105
IF-MIB::ifHCOutOctets.105 = Counter64: 42721
```

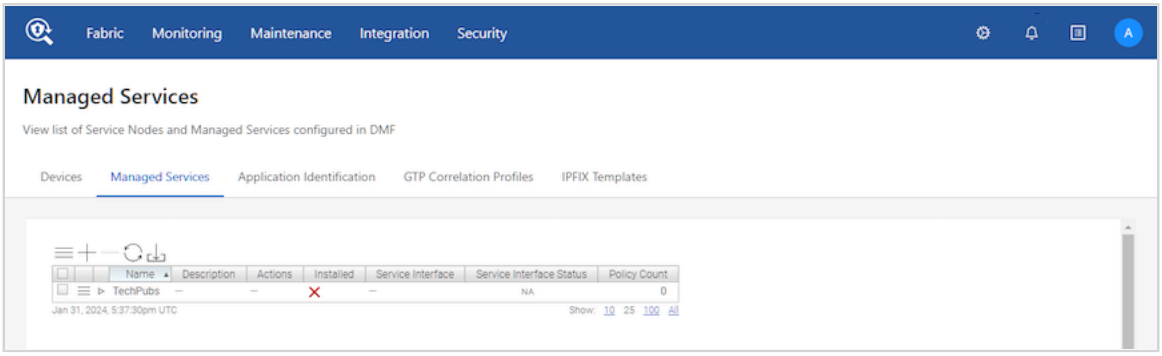
To fetch Link Up and Down status, enter the following command:

```
[root@TestTool anet]# snmpwalk -v2c -c onlonl 10.106.6.6 .1.3.6.1.2.1.2.2.1.8.109
IF-MIB::ifOperStatus.109 = INTEGER: down(2)
[root@TestTool anet]# snmpwalk -v2c -c onlonl 10.106.6.6 .1.3.6.1.2.1.2.2.1.8.105
IF-MIB::ifOperStatus.105 = INTEGER: up(1)
```

5.4 Configuring Managed Services

To view, edit, or create DANZ Monitoring Fabric (DMF) managed services, select the **Monitoring > Managed Services** option.

Figure 5-1: Managed Services



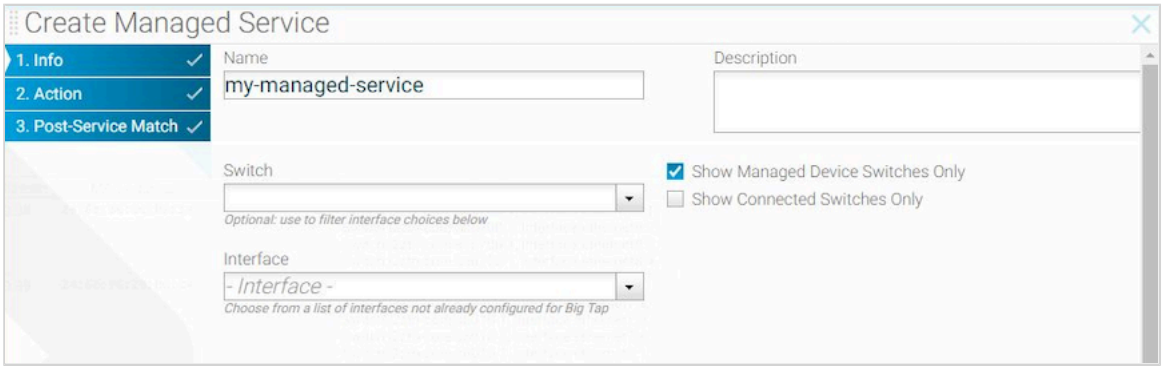
This page displays the service node appliance devices connected to the DMF Controller and the services configured on the Controller.

5.4.1 Using the GUI to Define a Managed Service

To create a new managed service, perform the following steps:

1. Select the **Provision** control (+) in the **Managed Services** table. The system displays the **Create Managed Service** dialog, shown in the following figure.

   Figure 5-2: Create Managed Service: Info



2. Assign a name to the managed service.
3. (Optional) Provide a text description of the managed service.
4. Select the switch and interface providing the service.

   The **Show Managed Device Switches Only** checkbox, enabled by default, limits the switch selection list to service node appliances. Enable the **Show Connected Switches Only** checkbox to limit the display to connected switches.
5. Select the action from the Action selection list, which provides the following options.
   - **Application ID**
   - **Deduplication**: Deduplicate selected traffic, including NATed traffic.
   - **GTP Correlation**
   - **Header Strip**: Remove bytes of packet starting from zero till selected Anchor and offset bytes
   - **Header Strip Cisco Fabric Path Header**: Remove the Cisco Fabric Path encapsulation header
   - **Header Strip ERSPAN Header**: Remove Encapsulated Remote Switch Port Analyzer Encapsulation header
   - **Header Strip Genev1 Header**: Remove Generic Network Virtualization Encapsulation header
   - **Header Strip L3 MPLS Header**: Remove Layer 3 MPLS encapsulation header
   - **Header Strip LISP Header**: Remove Locator Separation Protocol Encapsulation header
   - **Header Strip VXLAN Header**: Remove Virtual Extensible LAN Encapsulation header
   - **IPFIX**: Generate IPFIX by selecting matching traffic and forwarding it to specified collectors.
   - **Mask**: Mask sensitive information as specified by the user in packet fields.
   - **NetFlow**: Generate a NetFlow by selecting matching traffic and forwarding it to specified collectors.

- **Pattern-Drop**: Drop matching traffic.
- **Pattern Match**: Forward matching traffic.
- **Session Slice**: Slice TCP sessions.
- **Slice**: Slice the given number of bytes based on the specified starting point in the packet.
- **TCP Analysis**
- **Timestamp**: Identify the time that the service node receives the packet.
- **UDP Replication**: Copy UDP messages to multiple IP destinations, such as Syslog or NetFlow messages.

6. (Optional) Identify the starting point for service actions.

   Identify the start point for the deduplication, mask, pattern-match, pattern-drop services, or slice services using one of the keywords listed below.

   - **packet-start**: add the number of bytes specified by the ***integer*** value to the first byte in the packet.
   - **l3-header-start**: add the number of bytes specified by the ***integer*** value to the first byte in the Layer 3 header.
   - **l4-header-start**: add the number of bytes specified by the ***integer*** value to the first byte in the layer-4 header.
   - **l4-payload-start**: add the number of bytes specified by the ***integer*** value to the first byte in the layer-4 user data.
   - ***integer***: specify the number of bytes to offset for determining the start location for the service action relative to the specified start keyword.

7. To assign a managed service to a policy, enable the checkbox on the **Managed Services** page of the **Create Policy** or **Edit Policy** dialog.
8. Select the backup service from the Backup Service selection list to create a backup service. The backup service is used when the primary service is not available.

5.4.2 Using the CLI to Define a Managed Service

> **Note:** When connecting a LAG interface to the DANZ Monitoring Fabric (DMF) service node appliance, member links should be of the same speed and can span across multiple service nodes. The maximum number of supported member links per LAG interface is 32, which varies based on the switch platform. Please refer to the hardware guide for the exact details of the supported configuration.

To configure a service to direct traffic to a DMF service node, complete the following steps:

1. Define an identifier for the managed service by entering the following command:

```
controller-1(config)# managed-service DEDUPLICATE-1
controller-1(config-managed-srv)#
```

   This step enters the **config-managed-srv** submode to configure a DMF-managed service.

2. (Optional) Configure a description for the current managed service by entering the following command:

```
controller-1(config-managed-srv)# description "managed service for policy DEDUPLICATE-1"
```

   The following are the commands available from this submode:

   - **description**: provide a service description
   - **post-service-match**: select traffic after applying the header strip service
   - Action sequence number in the range [***1 - 20000***]: identifier of service action
   - **service-interface**: associate an interface with the service

3. Use a number in the range [***1 - 20000***] to identify a service action for a managed service.

   The following summarizes the available service actions. See the subsequent sections for details and examples for specific service actions.

   - **dedup** {**anchor-offset** | **full-packet** | **routed-packet**}
   - **header-strip** {**l4-header-start** | **l4-payload-start** | **packet-start** }[***offset***]
   - **decap-cisco-fp** {**drop**}
   - **decap-erspan** {**drop**}
   - **decap-geneve** {**drop**}
   - **decap-l3-mpls** {**drop**}
   - **decap-lisp** {**drop**}
   - **decap-vxlan** {**drop**}
   - **mask** {***mask***/***pattern***} [{**packet-start** | **l3-header-start** | **l4-header-start** | **l4-payload-start**} ***mask***/***offset***] [***mask***/***mask-start mask***/***mask-end***]}
   - **netflow** ***Delivery_interface Name***
   - **ipfix** ***Delivery_interface Name***

- **udp-replicate** ***Delivery_interface Name***
- **tcp-analysis** ***Delivery_interface Name***

> **Note:** The IPFIX, NetFlow, and udp-replicate service actions enable a separate submode for defining one or more specific configurations. One of these services must be the last service applied to the traffic selected by the policy.

- **pattern-drop** ***pattern*** [{**l3-header-start** | **l4-header-start** | **packet-start** }]
- **pattern-match** ***pattern*** [{**l3-header-start** | **l4-header-start** | **packet-start** }] |
- **slice** {**packet-start** | **l3-header-start** | **l4-header-start** | ***l4-payload-start***} ***integer***}
- **timestamp**

For example, the following command enables packet deduplication on the routed packet:

```
controller-1(config-managed-srv)# 1 dedup routed-packet
```

4. Optionally, identify the start point for the mask, pattern-match, pattern-drop services, or slice services.
5. Identify the service interface for the managed service by entering the following command:

```
controller-1(config-managed-srv)# service-interface switch DMF-CORE-SWITCH-1 ethernet40
```

Use a port channel instead of an interface to increase the bandwidth available to the managed service. The following example enables lag-interface1 for the service interface:

```
controller-1(config-managed-srv)# service-interface switch DMF-CORE-SWITCH-1 lag1
```

6. Apply the managed service within a policy like any other service, as shown in the following examples for deduplication, NetFlow, pattern matching (forwarding), and packet slicing services.

> **Note:** Multiple DMF policies can use the same managed service, for example, a packet slicing managed service.

5.4.3 Monitoring Managed Services

To identify managed services bound to a service node interface and the health status of the respective interface, use the following commands:

```
controller-1# show managed-service-device SN-Name interfaces
controller-1# show managed-service-device SN-Name stats
```

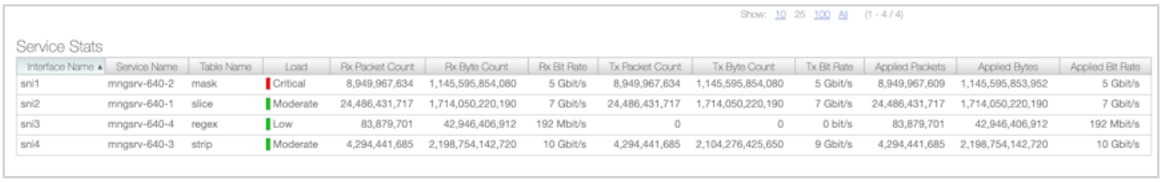
For example, the following command shows the managed services handled by the Service Node Interface (SNI):

```
controller-1#show managed-service-device <SN-Name> interfaces
------------------------------------ Interfaces ------------------------------------
#  Managed Service Interface Switch        Switch Interface Service Name
--|------------------------|------------|----------------|-------------------------|
1  sni1                    CORE-SWITCH-1 MS-LAG-1         MULTIPLE-SERVICES-ACTIVE-1
2  sni2                    CORE-SWITCH-1 MS-LAG-2         MULTIPLE-SERVICES-BACKUP-1
3  sni3                    CORE-SWITCH-1 MS-LAG-3         MULTIPLE-SERVICES-ACTIVE-2
4  sni4                    CORE-SWITCH-1 MS-LAG-4         MULTIPLE-SERVICES-BACKUP-2
5  sni5                    CORE-SWITCH-1 MS-LAG-5         MULTIPLE-SERVICES-ACTIVE-3
6  sni6                    CORE-SWITCH-1 MS-LAG-6         MULTIPLE-SERVICES-BACKUP-3
7  sni7                    CORE-SWITCH-1 MS-LAG-7
8  sni8                    CORE-SWITCH-1 MS-LAG-8
9  sni9                    CORE-SWITCH-1 MS-LAG-1         MULTIPLE-SERVICES-ACTIVE-1
10 sni10                   CORE-SWITCH-1 MS-LAG-2         MULTIPLE-SERVICES-BACKUP-1
-----------------------------------output truncated-----------------------------------
controller-1# show managed-service-device <SN-Name> stats
------------------------------------------------------------------------------------------------------------------------------------------------------------------------ Stats -------------------------------------------------------------------------------------------------------------------------
#  Device Int Name Service Name                Action  Load     Rx packets  Rx bytes       Rx Bit Rate Applied packets Applied bytes  App Bit Rate Tx packets  Tx bytes       Tx Bit Rate
--|------|--------|--------------------------|-------|--------|-----------|--------------|-----------|--------------|-------------|-----------|--------------|-----------|
1         sni1     MULTIPLE-SERVICES-ACTIVE-1 netflow moderate 388305411O2 225349518957T5 331Mbps     33491332356     193922807091Z7 316Mbps     64547484     2425572669Z6   4.77Mbps
2         sni1     MULTIPLE-SERVICES-ACTIVE-1 dedup   moderate 408181768O9 227626914950Z7 335Mbps     40790968622     227608511280S4 335Mbps     388033329S1  225331115394S1 331Mbps
3         sni3     MULTIPLE-SERVICES-ACTIVE-2 dedup   no       0                         0           -               0              0           -            0              0
4         sni3     MULTIPLE-SERVICES-ACTIVE-2 ipfix   no       0                         0           -               0              0           17048        1619560        48bps
5         sni5     MULTIPLE-SERVICES-ACTIVE-3 dapper  critical 14769229199 1070427935998 29.1Mbps    14769229193     1070427935998  29.1Mbps    1614349      133288798S5    367Kbps
6         sni5     MULTIPLE-SERVICES-ACTIVE-3 slice   critical 14769229199 871971235970S0 183Mbps    14762965759     871909743256Z  183Mbps     14769229199  1069801771998  29.1Mbps
7         sni9     MULTIPLE-SERVICES-ACTIVE-1 netflow low      387094162931 242037856637S4 132Mbps    33247678310     213322985651T4 116Mbps     64946609     242204089910   4.75Mbps
8         sni9     MULTIPLE-SERVICES-ACTIVE-1 dedup   low      39352939982
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------output truncated-------------------------------------------------------------------------------------------------------------------------
```

> **Note:** The **`show managed-service-device`** ***SN-Name*** **`stats`** ***Managed-service-name*** command filters the statistics of a specific managed service.

The Load column shows no, low, moderate, high, and critical health indicators. These health indicators are represented by green, yellow, and red under **DANZ Monitoring Fabric > Managed Services > Devices > Service Stats**. They reflect the processor load on the service node interface at that instant but do not show the bandwidth of the respective data port (SNI) handling traffic, as shown in the following sample snapshot of the **Service Stats** output.

Figure 5-3: Service Node Interface Load Indicator



5.5 Deduplication Action

The DANZ Monitoring Fabric (DMF) Service Node enhances the efficiency of network monitoring tools by eliminating duplicate packets. Duplicate packets can be introduced into the out-of-band monitoring data stream by receiving the same flow from multiple TAP or SPAN ports spread across the production network. Deduplication eliminates these duplicate packets and enables more efficient use of passive monitoring tools.

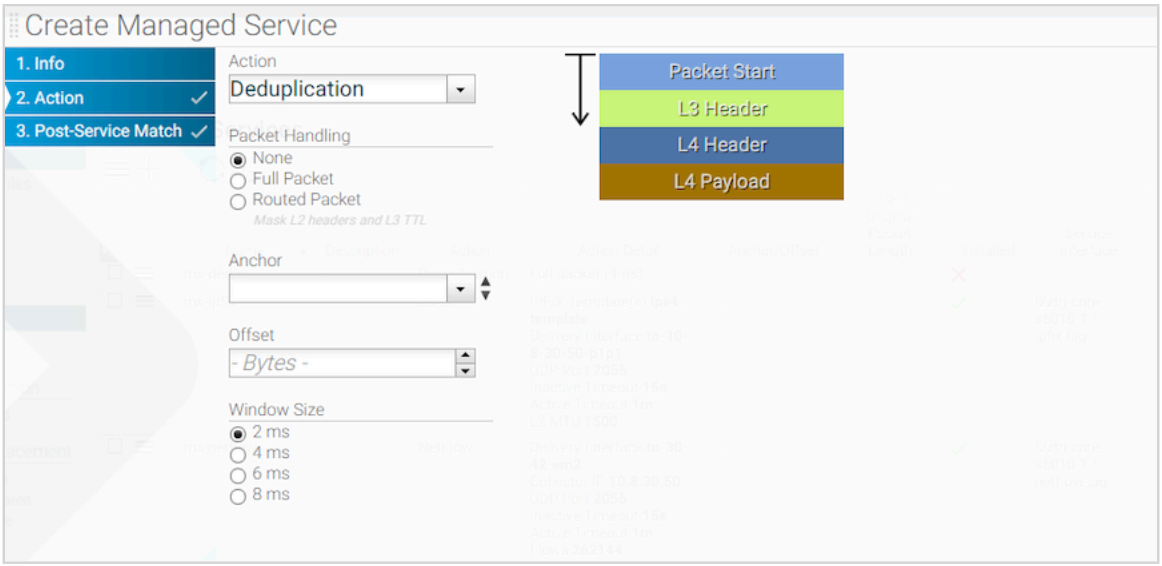
The DMF Service Node provides three modes of deduplication for different types of duplicate packets.

- Full packet deduplication: deduplicates incoming packets that are identical at the L2/L3/L4 layers.
- Routed packet deduplication: as packets traverse an IP network, the MAC address changes from hop to hop. Routed packet deduplication enables users to match packet contents starting from the L3 header.
- NATed packet deduplication: to perform NATed deduplication, the service node compares packets in the configured window that are identical starting from the L4 payload. To use NATed packet deduplication, perform the following fields as required:
  - Anchor: Packet Start, L2 Header Start, L3 Header Start, or L3 Payload Start fields.
  - Offset: the number of bytes from the anchor where the deduplication check begins.

The time window in which the service looks for duplicate packets is configurable. Select a value among these choices: ***2ms*** (the default), ***4ms***, ***6ms***, and ***8ms***.

GUI Configuration

Figure 5-4: Create Managed Service > Action: Deduplication Action



CLI Configuration

```
Controller-1(config)# show running-config managed-service MS-DEDUP-FULL-PACKET
! managed-service
managed-service MS-DEDUP-FULL-PACKET
description 'This is a service that does Full Packet Deduplication'
1 dedup full-packet window 8
service-interface switch CORE-SWITCH-1 ethernet13/1
Controller-1(config)#
```

```
Controller-1(config)# show running-config managed-service MS-DEDUP-ROUTED-PACKET
! managed-service
managed-service MS-DEDUP-ROUTED-PACKET
description 'This is a service that does Routed Packet Deduplication'
1 dedup routed-packet window 8
service-interface switch CORE-SWITCH-1 ethernet13/2
Controller-1(config)#
```

```
Controller-1(config)# show running-config managed-service MS-DEDUP-NATTED-PACKET
! managed-service
managed-service MS-DEDUP-NATTED-PACKET
description 'This is a service that does Natted Packet Deduplication'
1 dedup anchor-offset l4-payload-start 0 window 8
service-interface switch CORE-SWITCH-1 ethernet13/3
Controller-1(config)#
```

Note: The existing command is augmented to show the deduplication percentage. The command syntax is **show managed-service-device *Service-Node-Name* stats *dedup-service-name* dedup**.

```
Controller-1(config)# show managed-service-device DMF-SN-R740-1 stats MS-DEDUP dedup
~~~~~~~~~~~~~~~~ Stats ~~~~~~~~~~~~~~~~
Interface Name : sni16
Function : dedup
Service Name : MS-DEDUP
Rx packets : 9924950
Rx bytes : 4216466684
Rx Bit Rate : 1.40Gbps
Applied packets : 9923032
Applied bytes : 4216337540
Applied Bit Rate : 1.40Gbps
Tx packets : 9796381
Tx bytes : 4207106113
Tx Bit Rate : 1.39Gbps
Deduped frame count : 126651
Deduped percent : 1.2763336851075358
Load : low
Controller-1(config)#
```

5.6 Header Strip Action

This action removes specific headers from the traffic selected by the associated DANZ Monitoring Fabric (DMF) policy. Alternatively, define custom header stripping based on the starting position of the Layer-3 header, the Layer-4 header, the Layer-4 payload, or the first byte in the packet.

Use the following decap actions isolated from the header-strip configuration stanza:

- **decap-erspan**: remove the Encapsulated Remote Switch Port Analyzer (ERSPAN) header.
- **decap-cisco-fabric-path**: remove the Cisco FabricPath protocol header.
- **decap-l3-mpls**: remove the Layer-3 Multi-protocol Label Switching (MPLS) header.
- **decap-lisp**: remove the LISP header.
- **decap-vxlan** [**udp-port** ***vxlan port***]: remove the Virtual Extensible LAN (VXLAN) header.
- **decap-geneve**: remove the Geneve header.



Note: For the Header Strip and Decap actions, apply post-service rules to select traffic after stripping the original headers.

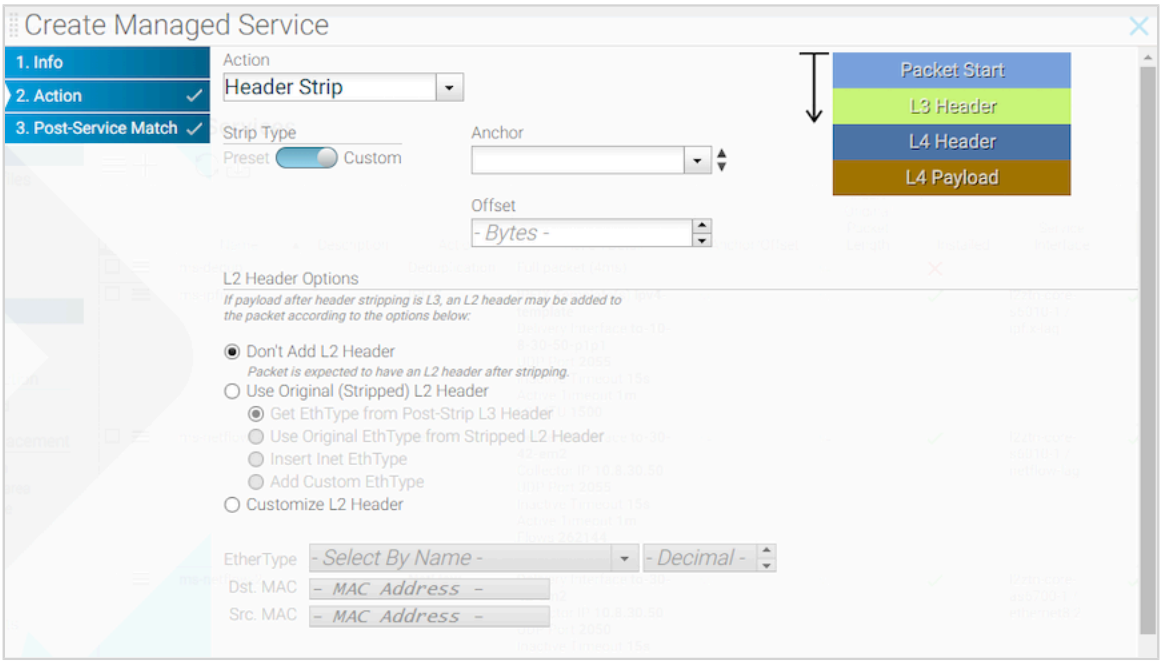
To customize the header-strip action, use one of the following keywords to strip up to the specified location in each packet:

- **l3-header-start**
- **l4-header-start**
- **l4-payload-start**
- **packet-start**

Input a positive integer representing the offset from which the strip action begins. When omitting an offset, the header stripping starts from the first byte in the packet.

GUI Configuration

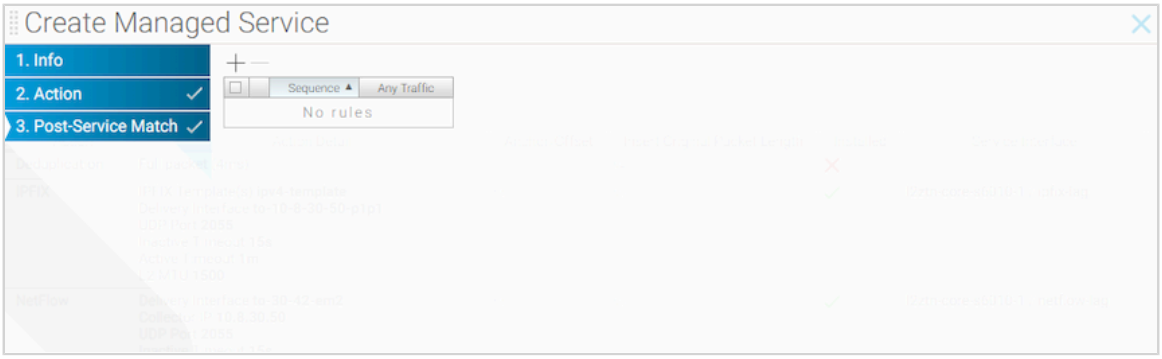
Figure 5-5: Create Managed Service: Header Strip Action



After assigning the required actions to the header stripping service, select **Next** or **Post-Service Match**.

The system displays the **Post Service Match** page, used in conjunction with the header strip service action.

Figure 5-6: Create Managed Service: Post Service Match for Header Strip Action



CLI Configuration

The header-strip service action strips the header and replaces it in one of the following ways:

- Add the original L2 **src-mac**, and **dst-mac**.
- Add the original L2 **src-mac**, **dst-mac**, and **ether-type**.
- Specify and add a custom **src-mac**, **dst-mac**, and **ether-type**.

The following are examples of custom header stripping:

This example strips the header and replaces it with the original L2 **src-mac** and **dst-mac**.

```
! managed-service
managed-service MS-HEADER-STRIP-1
1 header-strip packet-start 20 add-original-l2-dstmac-srcmac
service-interface switch CORE-SWITCH-1 ethernet13/1
```

This example adds the original L2 **src-mac**, **dst-mac**, and **ether-type**.

```
! managed-service
managed-service MS-HEADER-STRIP-2
1 header-strip packet-start 20 add-original-l2-dstmac-srcmac-ethertype
service-interface switch CORE-SWITCH-1 ethernet13/2
```

This example specifies the addition of a customized **src-mac**, **dst-mac**, and **ether-type**.

```
! managed-service
managed-service MS-HEADER-STRIP-3
1 header-strip packet-start 20 add-custom-l2-header 00:11:01:02:03:04 00:12:01:02:03:04
0x800
service-interface switch CORE-SWITCH-1 ethernet13/3
```

5.6.1 Configuring the Post-service Match

The post-service match configuration option enables matching on inner packet fields after the DANZ Monitoring Fabric (DMF) Service Node performs header stripping. This option is applied on the post-service interface after the service node completes the strip service action. Feature benefits include the following:

- The fabric can remain in L3/L4 mode. It is not necessary to change to offset match mode.
- Easier configuration.
- All match conditions are available for the inner packet.
- The policy requires only one managed service to perform the strip service action.

With this feature enabled, DMF knows exactly where to apply the post-service match. The following example illustrates this configuration.

```
! managed-service
managed-service MS-HEADER-STRIP-4
service-interface switch CORE-SWITCH-1 interface ethernet1
1 decap-l3-mpls
!
post-service-match
1 match ip src-ip 1.1.1.1
2 match tcp dst-ip 2.2.2.0 255.255.255.0
! policy
policy POLICY-1
filter-interface TAP-1
delivery-interface TOOL-1
use-managed-service MS-HEADER-STRIP-4 sequence 1
```

5.7 IPFIX and Netflow Actions

IP Flow Information Export (IP FIX), also known as NetFlow v10, is an IETF standard defined in ***RFC 7011***. The IPFIX generator (agent) gathers and transmits information about flows, which are sets of packets that contain all the keys specified by the IPFIX template. The generator observes the packets received in each flow and forwards the information to the IPFIX collector (server) in the form as a flowset.

Starting with the ***DANZ Monitoring Fabric (DMF)-7.1.0*** release, NetFlow v9 (Cisco proprietary) and IPFIX/ NetFlow v10 are both supported. Configuration of the IPFIX managed service is similar to configuration for earlier versions of NetFlow except for the UDP port definition. NetFlow v5 collectors typically listen over ***UDP port 2055***, while IFPIX collectors listen over ***UDP port 4739***.

NetFlow records are typically exported using User Datagram Protocol (UDP) and collected using a flow collector. For a NetFlow service, the service node takes incoming traffic and generates NetFlow records. The service node drops the original packets, and the generated flow records, containing metadata about each flow, are forwarded out of the service node interface.

5.7.1 IPFIX Template

The IPFIX template consists of the key element IDs representing IP flow, field element IDs representing actions the exporter has to perform over IP flows matching key element IDs, the template ID number for uniqueness, collector information, and eviction timers.

To define a template, configure keys of interest representing the IP flow and fields that identify the values measured by the exporter, the exporter information, and the eviction timers. To define the template, select the **Monitoring > Managed Service > IPFIX Template** option from the DANZ Monitoring Fabric (DMF) GUI or enter the **`ipfix-template`** ***`template-name`*** command in config mode, replacing ***template-name*** with a unique identifier for the template instance.

5.7.2 IPFIX Keys

Use an IPFIX key to specify the characteristics of the traffic to monitor, such as source and destination MAC or IP address, VLAN ID, Layer-4 port number, and QoS marking. The generator includes flows in a flow set having all the attributes specified by the keys in the template applied. The flowset is updated only for packets that have all the specified attributes. If a single key is missing, the packet is ignored. To see a listing of the keys supported in the current release of the DANZ Monitoring Fabric (DMF) Service Node, select the **Monitoring > Managed Service > IPFIX Template** option from the DMF GUI or type **`help key`** in ***config-ipxif-template*** submode.

The following are the keys supported in the current release:

Table 3: Supported Keys

destination-ipv4-address	**ip-version**
destination-ipv6-address	**policy-vlan-id**
destination-mac-address	**records-per-dmf-interface**
destination-transport-port	**source-ipv4-address**
dot1q-priority	**source-ipv6-address**
dot1q-vlan-id	**source-mac-address**
ethernet-type	**source-transport-port**
icmp-type-code-ipv4	**tcp-source-port** (introduced in DMF 8.8)
icmp-type-code-ipv6	**tcp-destination-port** (introduced in DMF 8.8)
ip-class-of-service	**udp-source-port** (introduced in DMF 8.8)
ip-diff-serv-code-point	**udp-destination-port** (introduced DMF 8.8)
ip-protocol-identifier	**vlan id**
ip-ttl	



Note: The **policy-vlan-id** and **records-per-dmf-interface** keys are Arista Proprietary Flow elements. The **policy-vlan-id** key helps to query per-policy flow information at Arista Analytics-node (Collector) in ***push-per-policy*** deployment mode. The **records-per-dmf-interface** key helps to identify filter interfaces tapping the traffic. The following limitations apply at the time of IPFIX template creation:

- The Controller will not allow the key combination of **source-mac-address** and **records-per-dmf-interface** in ***push-per-policy*** mode.
- The Controller will not allow the key combinations of **policy-vlan-id** and **records-per-dmf-interface** in ***push-per-filter*** mode.

5.7.3 IPFIX Fields

A field defines each value updated for the packets the generator receives that match the specified keys. For example, include fields in the template to record the number of packets, the largest and smallest packet sizes, or the start and end times of the flows. To see a listing of the fields supported in the current release of the DANZ Monitoring Fabric (DMF) Service Node, select the **Monitoring** > **Managed Service** > **IPFIX Template** option from the DMF GUI, or type **`help`** in ***config-ipxif-template*** submode. The following are the fields supported:

- ***flow-end-milliseconds***
- ***flow-end-reason***
- ***flow-end-seconds***
- ***flow-start-milliseconds***
- ***flow-start-seconds***
- ***maximum-ip-total-length***
- ***maximum-layer2-total-length***
- ***maximum-ttl***
- ***minimum-ip-total-length***
- ***minimum-layer2-total-length***
- ***minimum-ttl***
- ***octet-delta-count***
- ***packet-delta-count***
- ***tcp-control-bits***

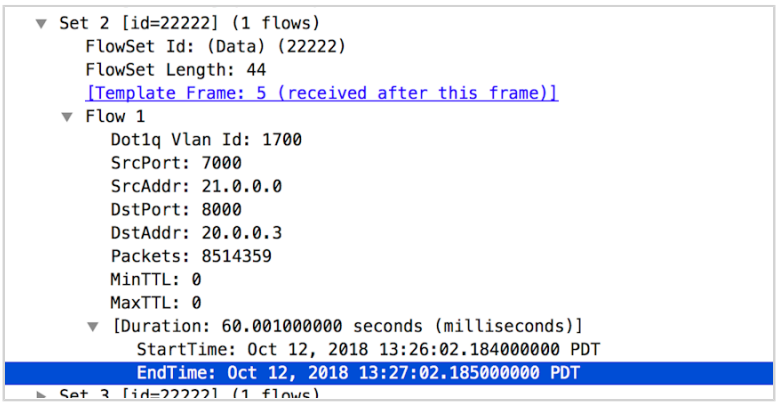
5.7.4 Active and Inactive Timers

After the number of minutes specified by the active timer, the flow set is closed and forwarded to the IPFIX collector. The default active timer is one minute. During the number of seconds set by the inactive timer, if no packets that match the flow definition are received, the flow set is closed and forwarded without waiting for the active timer to expire. The default value for the inactive time is ***15*** seconds.

5.7.5 Example Flowset

The following is a Wireshark view of an IPFIX flowset.

Figure 5-7: Example IPFIX Flowset in Wireshark



The following is a running-config that shows the IPFIX template used to generate this flowset.

Example IPFIX Template

```
! ipfix-template
ipfix-template Perf-temp
template-id 22222
key destination-ipv4-address
key destination-transport-port
key dot1q-vlan-id
key source-ipv4-address
key source-transport-port
field flow-end-milliseconds
field flow-end-reason
field flow-start-milliseconds
field maximum-ttl
field minimum-ttl
field packet-delta-count
```

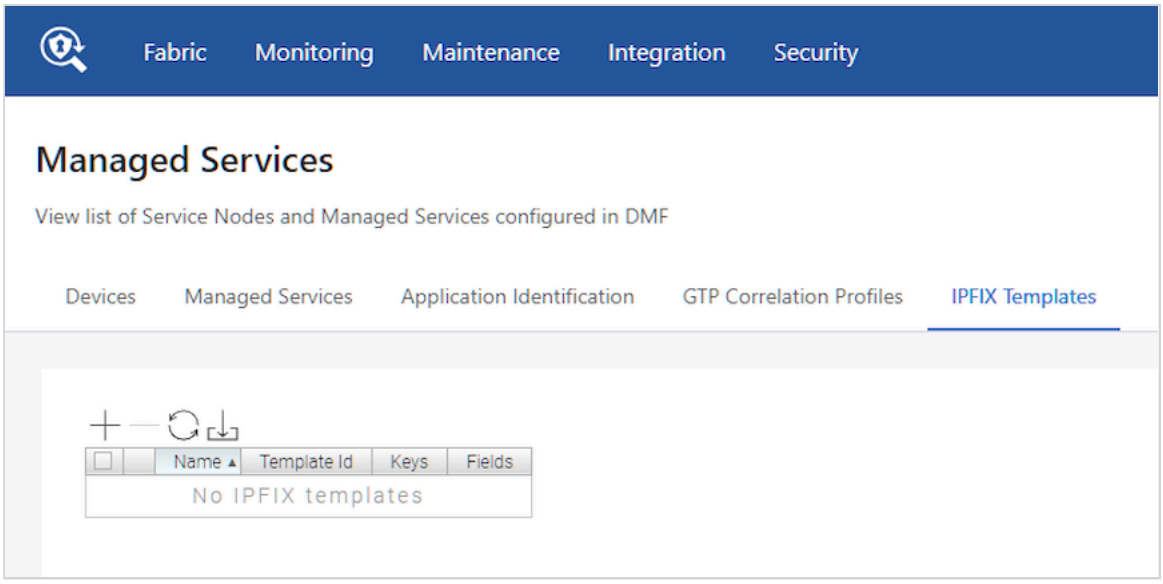
5.7.6 Using the GUI to Define an IPFIX Template

To define an IPFIX template, perform the following steps:

1. Select the **Monitoring > Managed Services** option.
2. On the **DMF Managed Services** page, select **IPFIX Templates**.

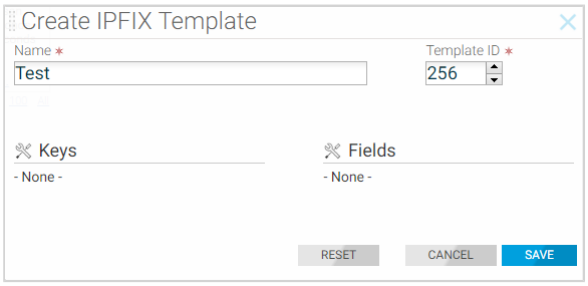
   The system displays the **IPFIX Templates** section.

   Figure 5-8: IPFIX Templates



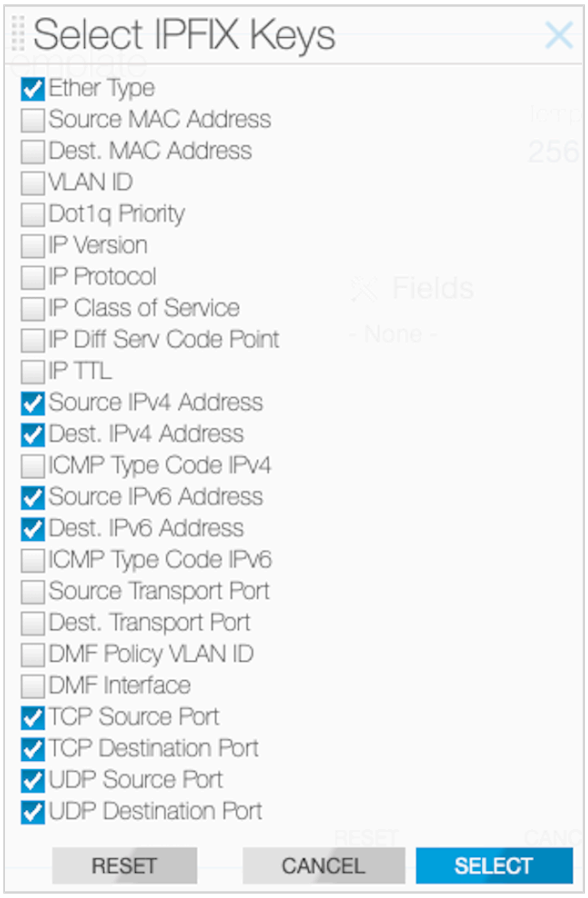
3. To create a new template, select the provision (**+**) icon in the **IPFIX Templates** section.

   Figure 5-9: Create IPFIX Template



4. To add an IPFIX key to the template, select the **Settings** control in the **Keys** section. The system displays the following dialog.

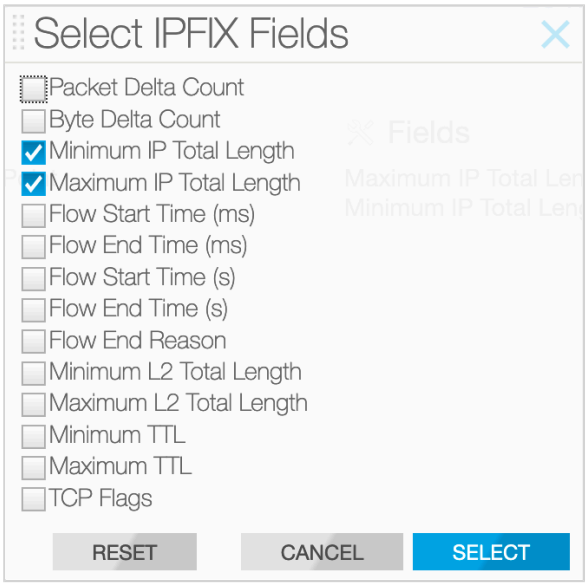
Figure 5-10: Select IPFIX Keys



5. Enable each checkbox for the keys to add to the template and use **Select**.

6. To add an IPFIX field to the template, select the **Settings** control in the **Fields** section. The system displays the following dialog:

Figure 5-11: Select IPFIX Fields



7. Enable the checkbox for each field to add to the template and use **Select**.
8. On the **Create IPFIX Template** page, select **Save**.

The new template is added to the **IPFIX Templates** table, with each key and field listed in the appropriate column. Use this customized template to apply when defining an IPFIX-managed service.

5.7.7 Using the CLI to Define an IPFIX Template

1. Create an IPFIX template.

```
controller-1(config)# ipfix-template IPFIX-IP
controller-1(config-ipfix-template)#
```

   This changes the CLI prompt to the ***config-ipfix-template*** submode.

2. Define the keys to use for the current template, using the following command:

   [**no**] **key** { **ethernet-type** | **source-mac-address** | **destination-mac-address** | **dot1q-vlan-id** | **dot1q-priority** | **ip-version** | **ip-protocol-identifier** | **ip-class-of-service** | **ip-diff-serv-code-point** | **ip-ttl** | **sourceipv4-address** | **destination-ipv4-address** | **icmp-type-code-ipv4** | **source-ipv6-address** | **destination-ipv6-address** | **icmp-type-code-ipv6** | **source-transport-port** | **destination-transport-port** }

   The keys specify the attributes of the flows to be included in the flowset measurements.

3. Define the fields to use for the current template, using the following command:

   [**no**] **field** { **packet-delta-count** | **octet-delta-count** | **minimum-ip-total-length** | **maximum-ip- total-length** | **flow-start-seconds** | **flow-end-seconds** | **flow-end-reason** | **flow-start-milliseconds** | **flow-end-milliseconds** | **minimum-layer2-total-length** | **maximum-layer2-total- length** | **minimum-ttl** | **maximum-ttl** }

   The fields specify the measurements to be included in the flowset.

4. Use the existing IPFIX template commands in the `config` mode to use the keys introduced in DMF 8.8.0.

   - **tcp-source-port**
   - **tcp-destination-port**
   - **udp-source-port**
   - **udp-destination-port**

```
controller-1(config)#
controller-1(config)# ipfix-template tcp-traffic
controller-1(config-ipfix-template)# key tcp-destination-port
controller-1(config-ipfix-template)# show this
! ipfix-template
ipfix-template tcp-traffic
key tcp-destination-port
```

   Use the same method for all keys added as part of this feature - `tcp-destination-port`, `tcp-source-port`, `udp-destination-port`, and `udp-source-port`.

Use the template when defining the IPFIX action.

Show Commands

The show commands specific to the new keys introduced in DMF 8.8.0 have not changed. If the new keys are used in any IPFIX template, they are displayed along with other keys.

View the keys used in a given IPFIX template using the following commands:

```
controller-1(config-ipfix-template)# show ipfix-template port-based-traffic
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Ipfix-templates ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Template Name      Keys                                                 Fields
-|------------------|----------------------------------------------------|------|
1 port-based-traffic ethernet-type, tcp-destination-port, tcp-source-port
```

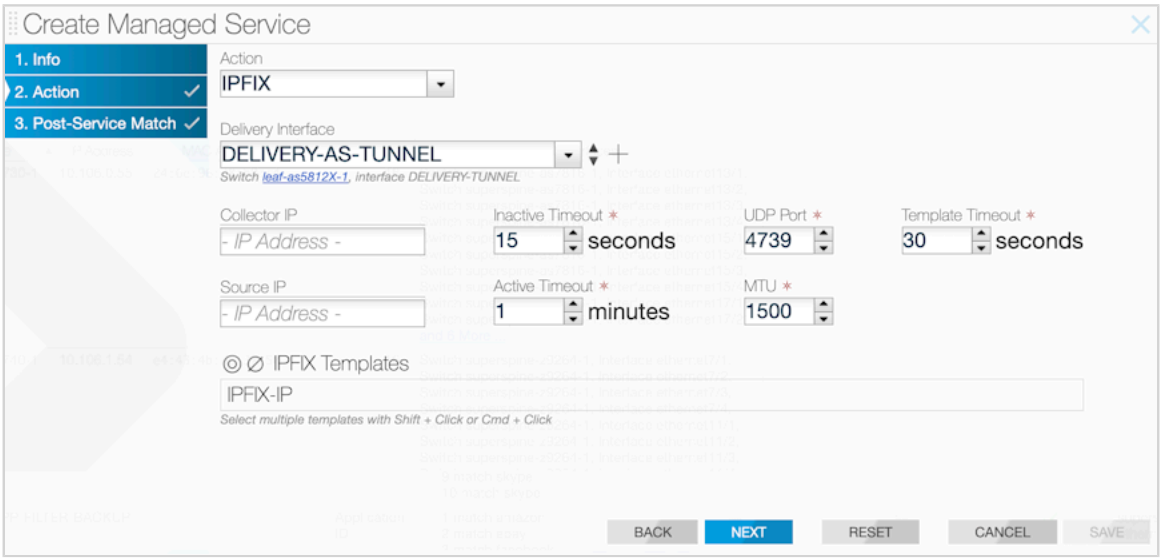
```
controller-1(config-ipfix-template)# show running-config ipfix-template port-based-traffic

! ipfix-template
ipfix-template port-based-traffic
  key ethernet-type
  key tcp-destination-port
  key tcp-source-port
```

5.7.8 Using the GUI to Define an IPFIX Service Action

Select IPFIX from the **Action** selection list on the **Create Managed Service > Action** page.

Figure 5-12: Selecting IPFIX Action in Create Managed Service



Enter the following required configuration details:

- Assign a delivery interface.
- Configure the collector IP address.
- Identify the IPFIX template.

The following configuration is optional:

- Inactive timeout: the interval of inactivity that marks a flow inactive.
- Active timeout: length of time between each IPFIX flows for a specific flow.
- Source IP: source address to use for the IPFIX flowsets.
- UDP port: UDP port to use for sending IPFIX flowsets.
- MTU: MTU to use for sending IPFIX flowsets.

After completing the configuration, select **Next**, and then select **Save**.

5.7.9 Using the CLI to Define an IPFIX Service Action

Define a managed service and define the IPFIX action.

```
controller(config)# managed-service MS-IPFIX-SERVICE
controller(config-managed-srv)# 1 ipfix TO-DELIVERY-INTERFACE
controller(config-managed-srv-ipfix)# collector 10.106.1.60
controller(config-managed-srv-ipfix)# template IPFIX-TEMPLATE
```

The **active-timeout** and **inactive-timeout** commands are optional

To view the running-config for a managed service using the IPFIX action, enter the following command:

```
controller1# show running-config managed-service MS-IPFIX-ACTIVE
! managed-service
managed-service MS-IPFIX-ACTIVE
service-interface switch CORE-SWITCH-1 ethernet13/1
!
1 ipfix TO-DELIVERY-INTERFACE
collector 10.106.1.60
template IPFIX-TEMPLATE
```

To view the IPFIX templates, enter the following command:

```
config# show running-config ipfix-template
! ipfix-template
ipfix-template IPFIX-IP
template-id 1974
key destination-ipv4-address
key destination-ipv6-address
key ethernet-type
key source-ipv4-address
key source-ipv6-address
field flow-end-milliseconds
field flow-end-reason
field flow-start-milliseconds
field minimum-ttl
field tcp-control-bits
------------------------output truncated------------------------
```

5.7.10 Records Per Interface Netflow using DST-MAC Rewrite

Destination MAC rewrite for the records-per-interface NetFlow and IPFIX feature is the default setting and applies to switches running Extensible Operating System (EOS) and SWL and is supported on all platforms.

A configuration option exists for using **src-mac** when overwriting the **dst-mac** isn't preferred.

Configurations using the CLI

Global Configuration

The global configuration is a central place to choose which rewrite option to use for the **records-per-interface**. The following example illustrates using **rewrite-src-mac** or **rewrite-dst-mac** in conjunction with the **filter-mac-rewrite** command.

```
c1(config)# filter-mac-rewrite rewrite-src-mac
c1(config)# filter-mac-rewrite rewrite-dst-mac
```

Netflow Configuration

The following example illustrates a NetFlow configuration.

```
c1(config)# managed-service ms1
c1(config-managed-srv)# 1 netflow
c1(config-managed-srv-netflow)# collector 213.1.1.20 udp-port 2055 mtu 1024 records-per-interface
```

IPFIX Configuration

The following example illustrates an IPFIX configuration.

```
c1(config)# ipfix-template i1
c1(config-ipfix-template)# field maximum-ttl
c1(config-ipfix-template)# key records-per-dmf-interface
c1(config-ipfix-template)# template-id 300

c1(config)# managed-service ms1
c1(config-managed-srv)# 1 ipfix
c1(config-managed-srv-ipfix)# template i1
```

Show Commands

NetFlow Show Commands

Use the **show running-config managed-service** command to view the NetFlow settings.

```
c1(config)# show running-config managed-service
! managed-service
managed-service ms1
  !
  1 netflow
    collector 213.1.1.20 udp-port 2055 mtu 1024 records-per-interface
```

IPFIX Show Commands

Use the **show ipfix-template i1** command to view the IPFIX settings.

```
c1(config)# show ipfix-template i1
~~~~~~~~~~~~~~~~~~ Ipfix-templates  ~~~~~~~~~~~~~~~~~~
# Template Name Keys                       Fields
-|-------------|------------------------|-----------|
1 i1            records-per-dmf-interface maximum-ttl

c1(config)# show running-config managed-service
! managed-service
managed-service ms1
  !
  1 ipfix
    template i1
```

Limitations

- The **filter-mac-rewrite rewrite-src-mac** command cannot be used on the **filter interface** that is part of the policy using **timestamping replace-src-mac**. However, the command works when using a **timestamping add-header-after-l2** configuration.

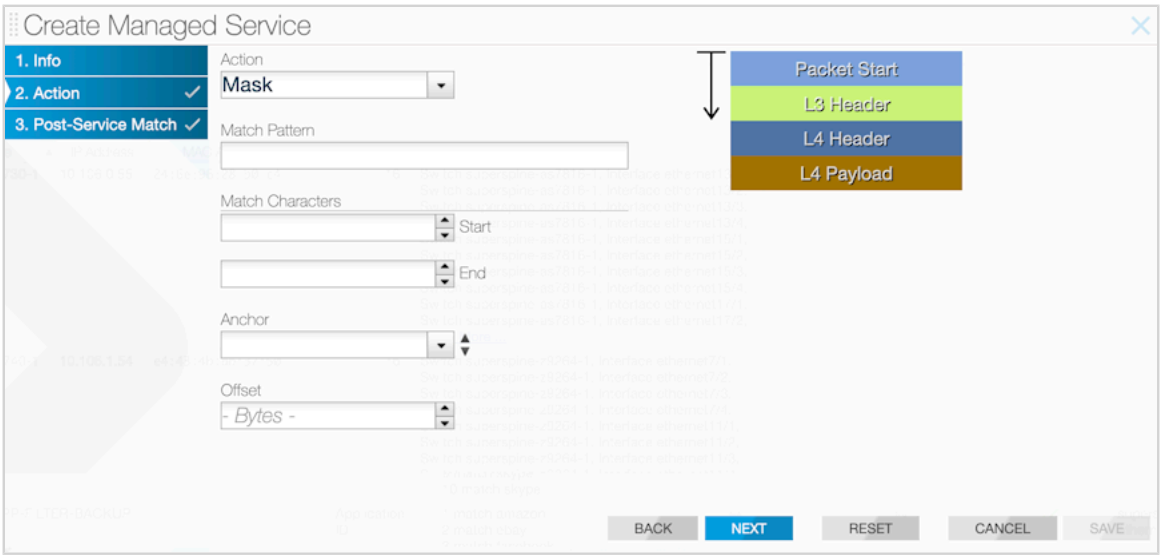
5.8 Packet-masking Action

The packet-masking action can hide specific characters in a packet, such as a password or credit card number, based on offsets from different anchors and by matching characters using regular (regex) expressions.

The mask service action applies the specified mask to the matched packet region.

GUI Configuration

Figure 5-13: Create Managed Service: Packet Masking



CLI Configuration

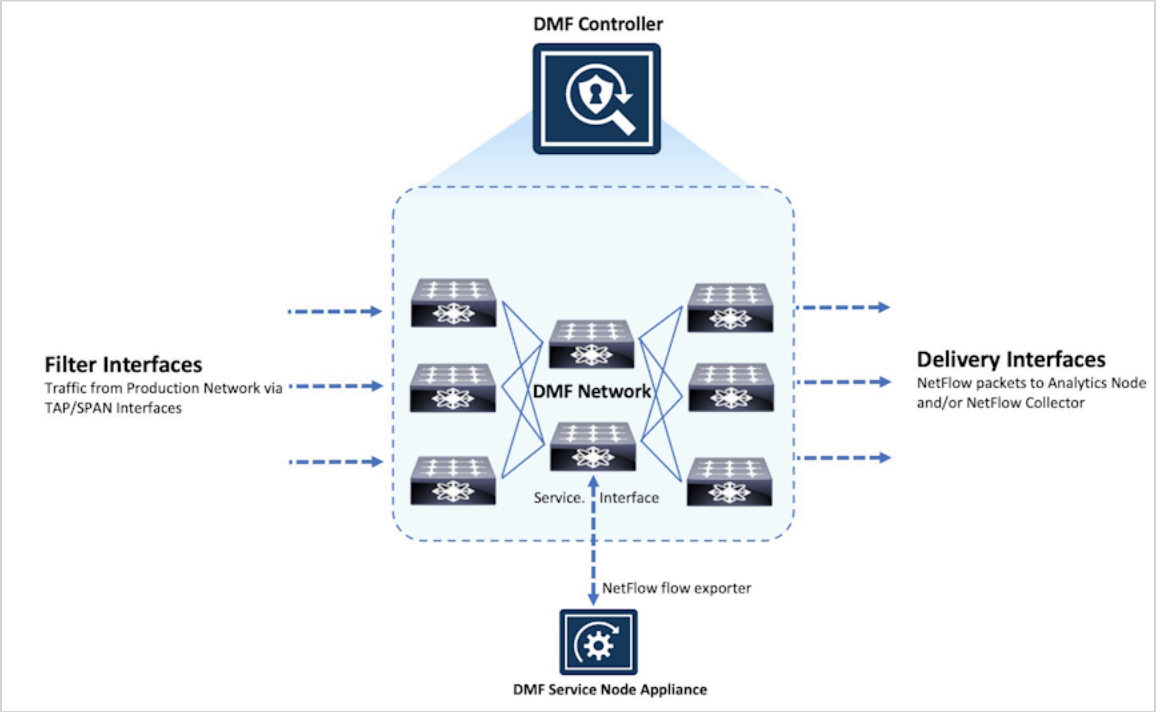
```
Controller-1(config)# show running-config managed-service MS-PACKET-MASK
! managed-service
managed-service MS-PACKET-MASK
description "This service masks pattern matching an email address in payload with X"
1 mask ([a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+.[a-zA-Z0-9_-]+)
service-interface switch CORE-SWITCH-1 ethernet13/1
```

5.9 Arista Analytics Node Capability

Arista Analytics Node capabilities are enhanced to handle NetFlow V5/V9 and IPFIX Packets. All these flow data are represented with the Netflow index.

Note: NetFlow flow record generation is enhanced for selecting VXLAN traffic. For VXLAN traffic, flow processing is based on inner headers, with the VNI as part of the key for flow lookup because IP addresses can overlap between VNIs.

Figure 5-14: NetFlow Managed Service



NetFlow records are exported using User Datagram Protocol (UDP) to one or more specified NetFlow collectors. Use the DMF Service Node to configure the NetFlow collector IP address and the destination UDP port. The default UDP port is ***2055***.

Note: No other service action, except the UDP replication service, can be applied after a NetFlow service action because part of the NetFlow action is to drop the packets.

5.9.1 Configuring the Arista Analytics Node Using the GUI

From the **Arista Analytics Node** dashboard, apply filter rules to display specific flow information.

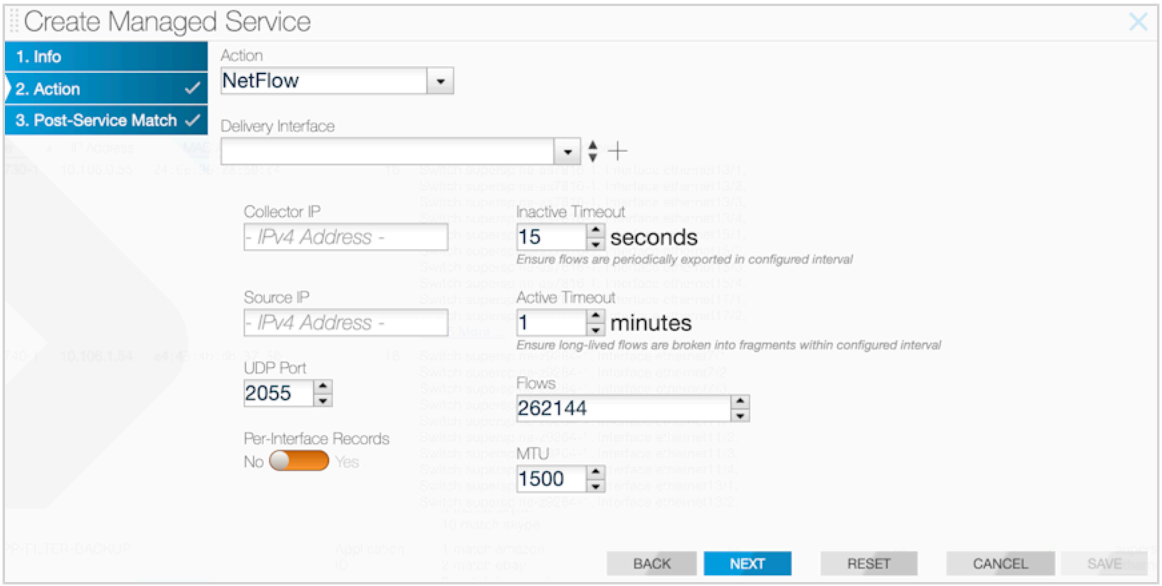
The following are the options available on this page:

- Delivery interface: interface to use for delivering NetFlow records to collectors.

> **Note:** The next-hop address must be resolved for the service to be active.

- Collector IP: identify the NetFlow collector IP address.
- Inactive timeout: use the **`inactive-timeout`** command to configure the interval of inactivity before NetFlow times out. The default is ***15*** seconds.
- Source IP: specify a source IP address to use as the source of the NetFlow packets.
- Active timeout: use active timeout to configure a period that a NetFlow can be generated continuously before it is automatically terminated. The default is one minute.
- UDP port: change the UDP port number used for the NetFlow packets. The default is ***2055***.
- Flows: specify the maximum number of NetFlow packets allowed. The allowed range is ***32768*** to ***1048576***. The default is ***262144***.
- Per-interface records: identify the filter interface where the NetFlow packets were originally received. This information can be used to identify the hop-by-hop path from the filter interface to the NetFlow collector.
- MTU: change the Maximum Transmission Unit (MTU) used for NetFlow packets.

Figure 5-15: Create Managed Service: NetFlow Action



5.9.2 Configuring the Arista Analytics Node Using the CLI

Use the **`show managed-services`** command to display the ARP resolution status.

> **Note:** The DANZ Monitoring Fabric (DMF) Controller resolves ARP messages for each NetFlow collector IP address on the delivery interface that matches the defined subnet. The subnets defined on the delivery interfaces cannot overlap and must be unique for each delivery interface.

Enter the **`1 netflow`** command and identify the configuration name and the submode changes to the ***config-managed-srv-netflow*** mode for viewing and configuring a specific NetFlow configuration.

The DMF Service Node replicates NetFlow packets received without changing the source IP address. Packets that do not match the specified destination IP address and packets that are not IPv4 or UDP are passed through. To configure a NetFlow-managed service, perform the following steps:

1. Configure the IP address on the delivery interface.

   This IP address is the next-hop IP address from the DANZ Monitoring Fabric towards the NetFlow collector.

   ```
   CONTROLLER-1(config)# switch DMF-DELIVERY-SWITCH-1
   CONTROLLER-1(config-switch)# interface ethernet1
   CONTROLLER-1(config-switch-if)# role delivery interface-name NETFLOW-DELIVERY-PORT ip-address
    172.43.75.1 nexthop-ip 172.43.75.2 255.255.255.252
   ```

2. Configure the rate-limit for the NetFlow delivery interface.

   ```
   CONTROLLER-1(config)# switch DMF-DELIVERY-SWITCH-1
   CONTROLLER-1(config-switch)# interface ethernet1
   CONTROLLER-1(config-switch-if)# role delivery interface-name NETFLOW-DELIVERY-PORT ip-address
    172.43.75.1 nexthop-ip 172.43.75.2 255.255.255.252
   CONTROLLER-1(config-switch-if)# rate-limit 256000
   ```

   > **Note:** The rate limit must be configured when enabling Netflow. When upgrading from a version of DMF before release ***6.3.1***, the Netflow configuration is not applied until a rate limit is applied to the delivery interface.

3. Configure the NetFlow managed service using the **`1 netflow`** command followed by an identifier for the specific NetFlow configuration.

   ```
   CONTROLLER-1(config)# managed-service MS-NETFLOW-SERVICE CONTROLLER-1
   (config-managed-srv)# 1 netflow NETFLOW-DELIVERY-PORT CONTROLLER-1
   (config-managed-srv-netflow)#
   ```

   The following commands are available in this submode:

   - **`active-timeout`**: configure the maximum length of time the NetFlow is transmitted before it is ended (in minutes).
   - **`collector`**: configure the collector IP address, and change the UDP port number or the MTU.
   - **`inactive-timeout`**: configure the length of time that the NetFlow is inactive before it is ended (in seconds).
   - **`max-flows`**: configure the maximum number of flows managed.

   An option exists to limit the number of flows or change the inactivity timeout using the max-flows or active timeout, or inactive timeout commands.

4. Configure the NetFlow collector IP address using the following command:

```
collector ip4-address[udp-port integer][mtu integer][records-per-interface]
```

The IP address, in IPV4 dotted-decimal notation, is required. The MTU and UDP port are required when changing these parameters from the defaults. Enable the **records-per-interface** option to allow identification of the filter interfaces from which the Netflow originated. Configure the Arista Analytics Node to display this information, as described in the ***DMF User Guide***.

The following example illustrates changing the Netflow UDPF port to ***9991***.

```
collector 10.181.19.31 udp-port 9991
```



Note: The IP address must be in the same subnet as the configured next hop and unique. It cannot be the same as the Controller, service node, or any monitoring fabric switch IP address.

5. Configure the DMF policy with the forward action and add the managed service to the policy.



Note: A DMF policy does not require any configuration related to a delivery interface for NetFlow policies because the DMF Controller automatically assigns the delivery interface.

The example below shows the configuration required to implement two NetFlow service instances (MS-NETFLOW-1 and MS-NETFLOW-1).

```
! switch
switch DMF-DELIVERY-SWITCH-1
!
interface ethernet1
role delivery interface-name NETFLOW-DELIVERY-PORT-1 ip-address 10.3.1.1
nexthop-ip 10.3.1.2 255.255.255.0
interface ethernet2
role delivery interface-name NETFLOW-DELIVERY-PORT-2 ip-address 10.3.2.1
nexthop-ip 10.3.2.2 255.255.255.0
! managed-service
managed-service MS-NETFLOW-1
service-interface switch DMF-CORE-SWITCH-1 interface ethernet11/1
!
1 netflow NETFLOW-DELIVERY-PORT-1
collector-ip 10.106.1.60 udp-port 2055 mtu 1024
managed-service MS-NETFLOW-2
service-interface switch DMF-CORE-SWITCH-2 interface ethernet12/1
!
1 netflow NETFLOW-DELIVERY-PORT-1
collector-ip 10.106.2.60 udp-port 2055 mtu 1024
! policy
policy GENERATE-NETFLOW-1
action forward
filter-interface TAP-INTF-DC1-1
filter-interface TAP-INTF-DC1-2
use-managed-service MS-NETFLOW-1 sequence 1
1 match any
policy GENERATE-NETFLOW-2
action forward
filter-interface TAP-INTF-DC2-1
filter-interface TAP-INTF-DC2-2
use-managed-service MS-NETFLOW-2 sequence 1
1 match any
```

5.10 Pattern-drop Action

The pattern-drop service action drops matching traffic.

Pattern matching allows content-based filtering beyond Layer-2, Layer-3, or Layer-4 Headers. This functionality allows filtering on the following packet fields and values:

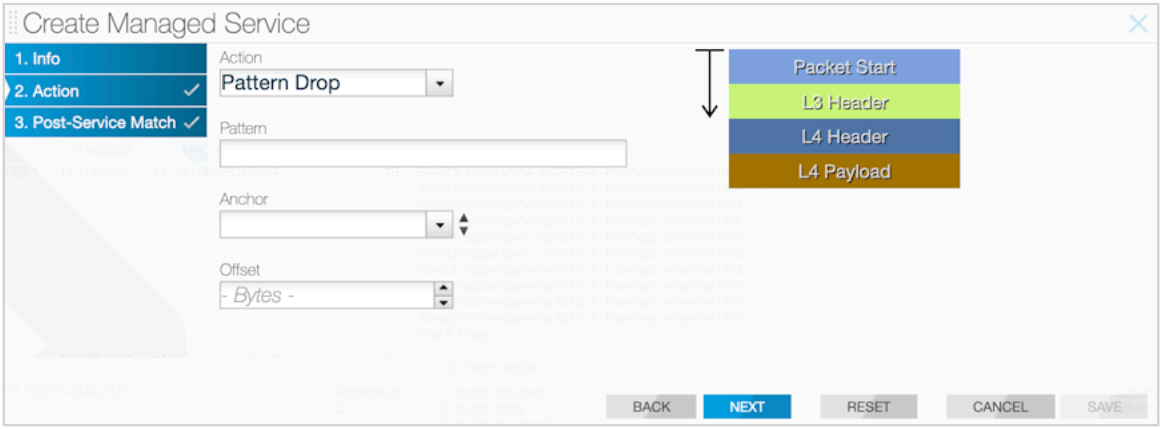
- URLs and user agents in the HTTP header
- patterns in BitTorrent packets
- encapsulation headers for specific parameters, including GTP, VXLAN, and VN-Tag
- subscriber device IP (user-endpoint IP)

Pattern matching allows Session-aware Adaptive Packet Filtering (SAPF) to identify HTTPS transactions on non-standard SSL ports. It can filter custom applications and separate control traffic from user data traffic.

Pattern matching is also helpful in enforcing IT policies, such as identifying hosts using unsupported operating systems or dropping unsupported traffic. For example, the Windows OS version can be identified and filtered based on the user-agent field in the HTTP header. The user-agent field may appear at variable offsets, so a regular expression search is used to identify the specified value wherever it occurs in the packet.

GUI Configuration

Figure 5-16: Create Managed Service: Pattern Drop Action



CLI Configuration

```
Controller-1(config)# show running-config managed-service MS-PACKET-MASK
! managed-service
managed-service MS-PACKET-MASK
description "This service drops traffic that has an email address in its payload"
1 pattern-drop ([a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+.[a-zA-Z0-9_-]+)
service-interface switch CORE-SWITCH-1 ethernet13/1
```

5.11 Pattern-match Action

The pattern-match service action matches and forwards matching traffic and is similar to the pattern-drop service action.

Pattern matching allows content-based filtering beyond Layer-2, Layer-3, or Layer-4 Headers. This functionality allows filtering on the following packet fields and values:

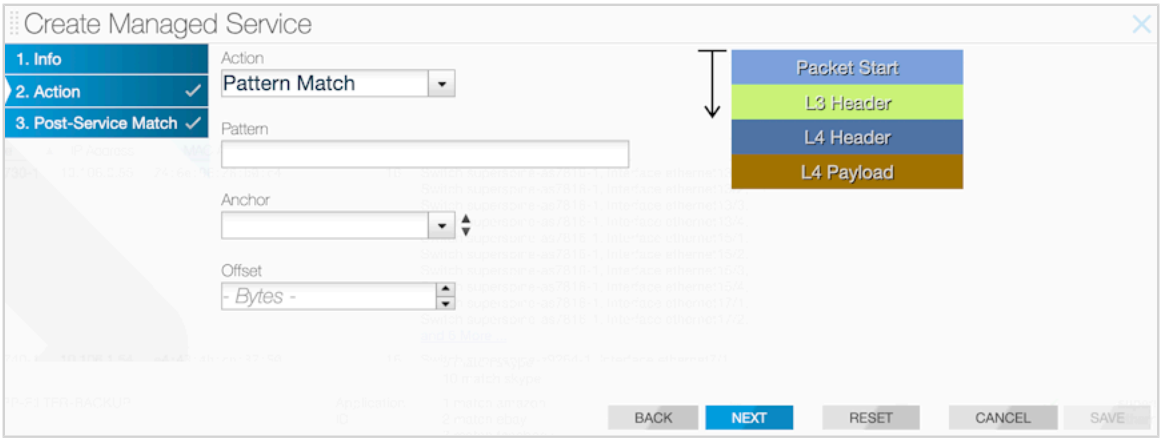
- URLs and user agents in the HTTP header
- patterns in BitTorrent packets
- encapsulation headers for specific parameters including, GTP, VXLAN, and VN-Tag
- subscriber device IP (user-endpoint IP)
- Pattern matching allows Session Aware Adaptive Packet Filtering and can identify HTTPS transactions on non-standard SSL ports. It can filter custom applications and can separate control traffic from user data traffic.

Pattern matching allows Session-aware Adaptive Packet Filtering (SAPF) to identify HTTPS transactions on non-standard SSL ports. It can filter custom applications and separate control traffic from user data traffic.

Pattern matching is also helpful in enforcing IT policies, such as identifying hosts using unsupported operating systems or dropping unsupported traffic. For example, the Windows OS version can be identified and filtered based on the user-agent field in the HTTP header. The user-agent field may appear at variable offsets, so a regular expression search is used to identify the specified value wherever it occurs in the packet.

GUI Configuration

Figure 5-17: Create Managed Service: Pattern Match Action



CLI Configuration

Use the **pattern-match *pattern*** keyword to enable the pattern-matching service action. Specify the pattern to match for packets to submit to the packet slicing operation.

The following example matches traffic with the string **`Windows NT 5.(0-1)`** anywhere in the packet and delivers the packets to the delivery interface ***TOOL-PORT-TO-WIRESHARK-1***. This service is optional and is applied to TCP traffic to destination ***port 80***.

```
! managed-service
managed-service MS-PATTERN-MATCH
description 'regular expression filtering'
1 pattern-match 'Windows\\sNT\\s5\\.[0-1]'
service-interface switch CORE-SWITCH-1 ethernet13/1
! policy
policy PATTERN-MATCH
action forward
delivery-interface TOOL-PORT-TO-WIRESHARK-1
description 'match regular expression pattern'
filter-interface TAP-INTF-FROM-PRODUCTION
priority 100
use-managed-service MS-PATTERN-MATCH sequence 1 optional
1 match tcp dst-port 80
```

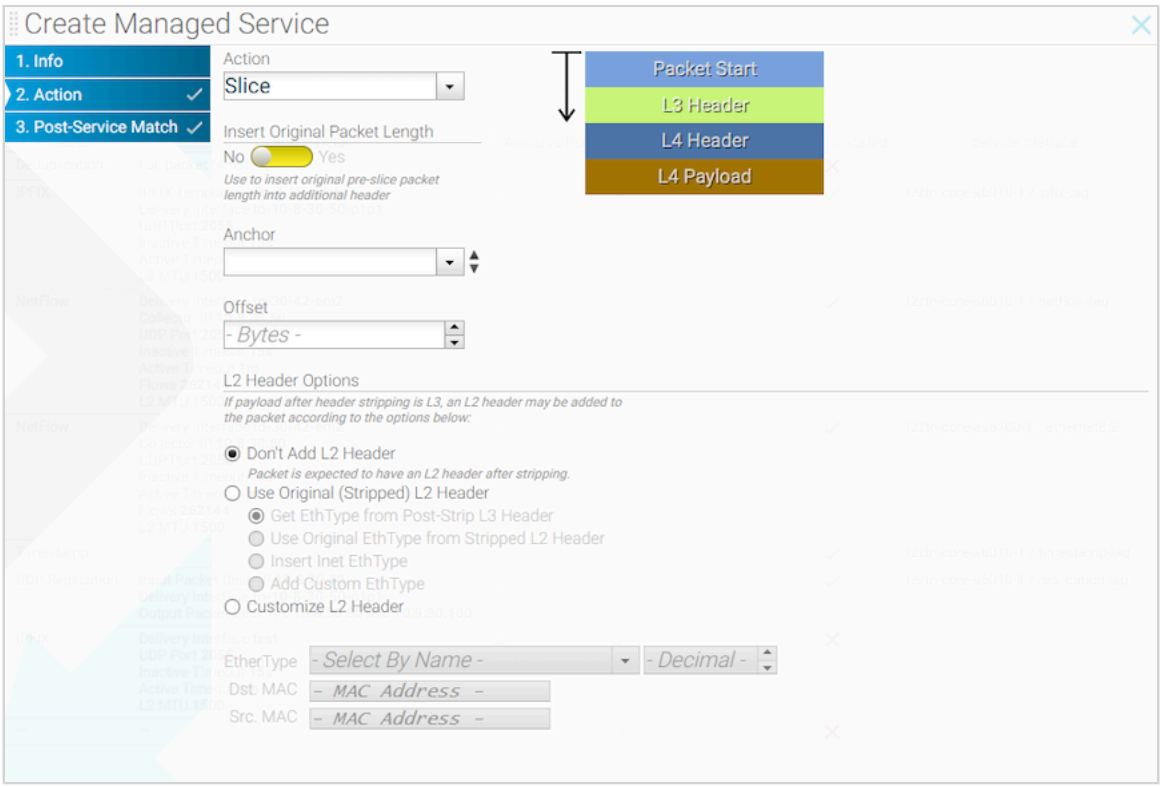
5.12 Slice Action

The slice service action slices the given number of packets based on the specified starting point in the packet. Packet slicing reduces packet size to increase processing and monitoring throughput. Passive monitoring tools process fewer bits while maintaining each packet's vital, relevant portions. Packet slicing can significantly increase the capacity of forensic recording tools. Apply packet slicing by specifying the number of bytes to forward based on an offset from the following locations in the packet:

- Packet start
- L3 header start
- L4 header start
- L4 payload start

GUI Configuration

Figure 5-18: Create Managed Service: Slice Action



This page allows inserting an additional header containing the original header length.

CLI Configuration

Use the **slice** keyword to enable the packet slicing service action and insert an additional header containing the original header length, as shown in the following example:

```
! managed-service
managed-service my-service-name
1 slice l3-header-start 20 insert-original-packet-length
```

```
service-interface switch DMF-CORE-SWITCH-1 ethernet20/1
```

The following example truncates the packet from the first byte of the Layer-4 payload, preserving just the original Ethernet header. The service is optional and is applied to all TCP traffic from ***port 80*** with the destination IP address ***10.2.19.119***

```
! managed-service
managed-service MS-SLICE-1
description 'slicing service'
1 slice l4-payload-start 1
service-interface switch DMF-CORE-SWITCH-1 ethernet40/1
! policy
policy slicing-policy
action forward
delivery-interface TOOL-PORT-TO-WIRESHARK-1
description 'remove payload'
filter-interface TAP-INTF-FROM-PRODUCTION
priority 100
use-managed-service MS-SLICE-1 sequence 1 optional
1 match tcp dst-ip 10.2.19.119 255.255.255.255 src-port 80
```

.

5.13 Packet Slicing on the 7280 Switch

This feature removes unwanted or unneeded bytes from a packet at a configurable byte position (offset). This approach is beneficial when the data of interest is situated within the headers or early in the packet payload. This action reduces the volume of the monitoring stream, particularly in cases where payload data is not necessary.

Another use case for packet slicing (slice action) can be removing payload data to ensure compliance with the captured traffic.

Within the DANZ Monitoring Fabric (DMF) fabric, two types of slice-managed services (packet slicing service) now exist. These types are distinguished based on whether installing the service on a service node or on an interface of a supported switch. The scope of this document is limited to the slice-managed service configured on a switch. The managed service interface is the switch interface used to configure this service.

All DMF 8.4 and above compatible 7280 switches support this feature. Use the **`show switch all property`** command to check which switch in DMF fabric supports this feature. The feature is supported if the **`Min Truncate Offset`** and **`Max Truncate Offset`** properties have a non-zero value.

```
# show switch all property
# Switch Min Truncate Offset  ...  Max Truncate Offset
-|------|-------------------| ... |---------------------------------
1 7280   100                 ...  9236
2 core1                      ...
```



Note: The CLI output example above is truncated for illustrative purposes. The actual output will differ.

5.13.1 Using the CLI to Configure Packet Slicing - 7280 Switch

Configure a **slice-managed** service on a switch using the following steps.

1. Create a managed service using the **managed-service *service name*** command.
2. Add **slice** action with **packet-start** anchor and an ***offset*** value between the supported range as reported by the **show switch all property** command.
3. Configure the service interface under the **config-managed-srv** submode using the service-interface switch ***switch-name interface-name*** command as shown in the following example.

```
> enable
# config
(config)# managed-service slice-action-7280-J2-J2C
(config-managed-srv)# 1 slice packet-start 101
(config-managed-srv)# service-interface switch 7280-J2-J2C Ethernet10/1
```

This feature requires the service interface to be in MAC loopback mode.

4. To set the service interface in MAC loopback mode, navigate to the **config-switch-if** submode and configure using the **loopback-mode mac** command, as shown in the following example.

```
(config)# switch 7280-J2-J2C
(config-switch)# interface Ethernet10/1
(config-switch-if)# loopback-mode mac
```

Once a managed service for slice action exists, any policy can use it.

5. Enter the **config-policy** submode, and chain the managed service using the **use-managed-service *service same* sequence *sequence*** command.

```
(config)# policy timestamping-policy
(config-policy)# use-managed-service slice-action-7280-J2-J2C sequence 1
```

Key points to consider while configuring the **slice** action on a supported switch:

1. Only the **packet-start** anchor is supported.
2. Ensure the offset is within the Min/Max truncate size bounds reported by the **show switch all property** command. If the configured value is beyond the bound, then DMF chooses the closest value of the range.

   For example, if a user configures the offset as 64, and the min truncate offset reported by switch properties is 100, then the offset used is 100. If the configured offset is 10,000 and the max truncate offset reported by the switch properties is 9236, then the offset used is 9236.
3. A configured offset for slice-managed service includes FCS when programmed on a switch interface, which means an offset of 100 will result in a packet size of 96 bytes (accounting for 4-byte FCS).
4. Configuring an offset below 17 is not allowed.
5. The same service interface cannot chain multiple managed services.
6. The **insert-original-packet-length** option is not applicable for switch-based slice-managed service.

CLI Show Commands

Use the **show policy *policy name*** command to see the runtime state of a policy using the slice-managed service. The command shows the service interface information and stats.

```
Controller# show policy packet-slicing-policy
Policy Name            : packet-slicing-policy
Config Status          : active - forward
Runtime Status         : installed
Detailed Status        : installed - installed to forward
Priority               : 100
Overlap Priority       : 0
# of switches with filter interfaces   : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces  : 1
# of filter interfaces : 1
# of delivery interfaces              : 1
# of core interfaces   : 0
# of services          : 1
# of pre service interfaces            : 1
# of post service interfaces           : 1
Push VLAN              : 1
Post Match Filter Traffic              : -
Total Delivery Rate    : -
Total Pre Service Rate : -
Total Post Service Rate: -
Overlapping Policies   : none
Component Policies     : none
Runtime Service Names  : packet-slicing-7280
Installed Time         : 2023-08-09 19:00:40 UTC
Installed Duration     : 1 hour, 17 minutes
~ Match Rules ~
# Rule
-|-----------|
1 1 match any

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF Switch IF Name     State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time
-|------|------|-----------|-----|---|-------|-----|--------|--------|------------------------------|
1 f1     7280   Ethernet2/1 up    rx  0       0     0        -        2023-08-09 19:00:40.305000 UTC

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF Switch IF Name     State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time
-|------|------|-----------|-----|---|-------|-----|--------|--------|------------------------------|
1 d1     7280   Ethernet3/1 up    tx  0       0     0        -        2023-08-09 19:00:40.306000 UTC

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Service Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Service name        Role Switch IF Name      State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time
-|------------------|----|------|------------|-----|---|-------|-----|--------|--------|------------------------------|
1 packet-slicing-7280 pre  7280   Ethernet10/1 up    tx  0       0     0        -        2023-08-09 19:00:40.305000 UTC
2 packet-slicing-7280 post 7280   Ethernet10/1 up    rx  0       0     0        -        2023-08-09 19:00:40.306000 UTC

~ Core Interface(s) ~
None.

~ Failed Path(s) ~
None.
```

Use the **show managed-services** command to view the status of all the managed services, including the packet-slicing managed service on a switch.

```
Controller# show managed-services
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Managed-services ~~~~~~~~~~~~~~~~~~~
```

5.13.2 Using the GUI to Configure Packet Slicing - 7820 Switch

Perform the following steps to configure or edit a managed service.

Managed Service Configuration

1. To configure or edit a managed service, navigate to the **DMF Managed Services** page from the **Monitoring** menu and select **Managed Services**.

Figure 5-19: DANZ Monitoring Fabric (DMF) Managed Services

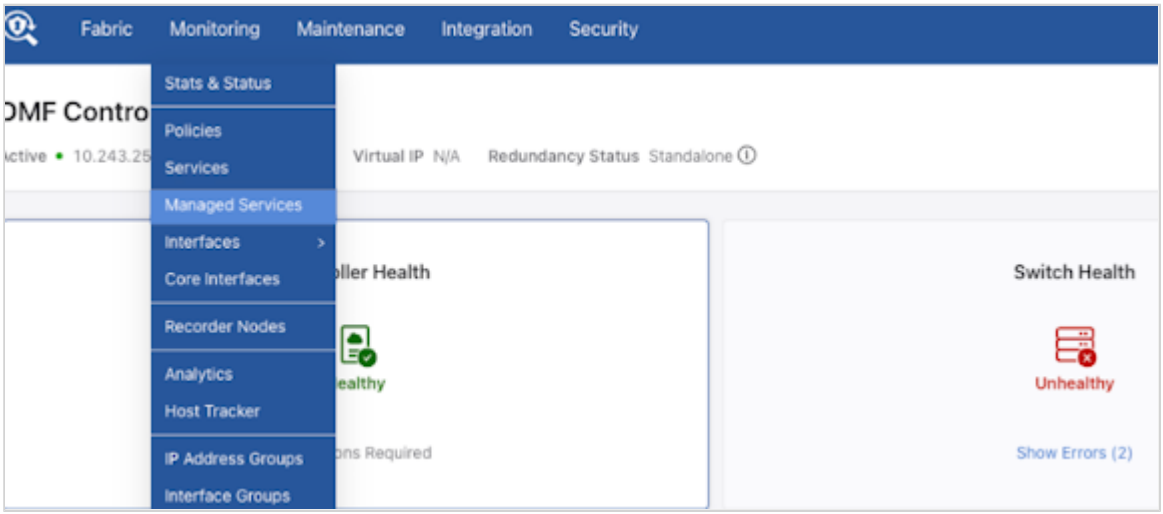
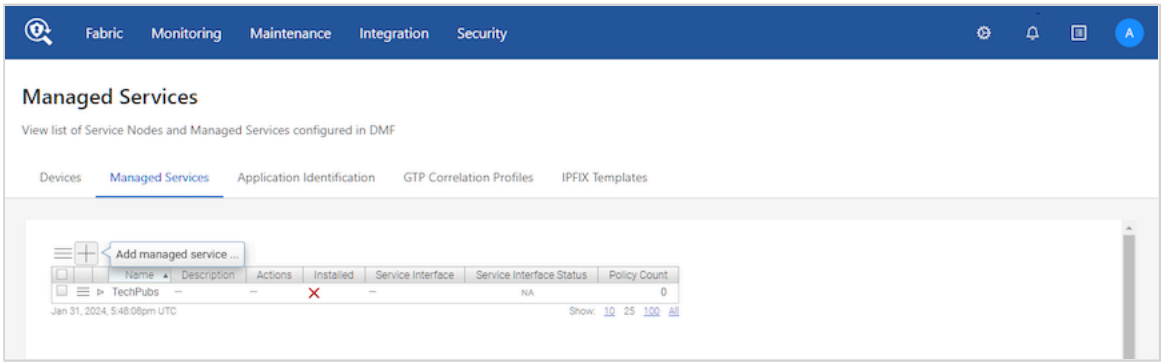
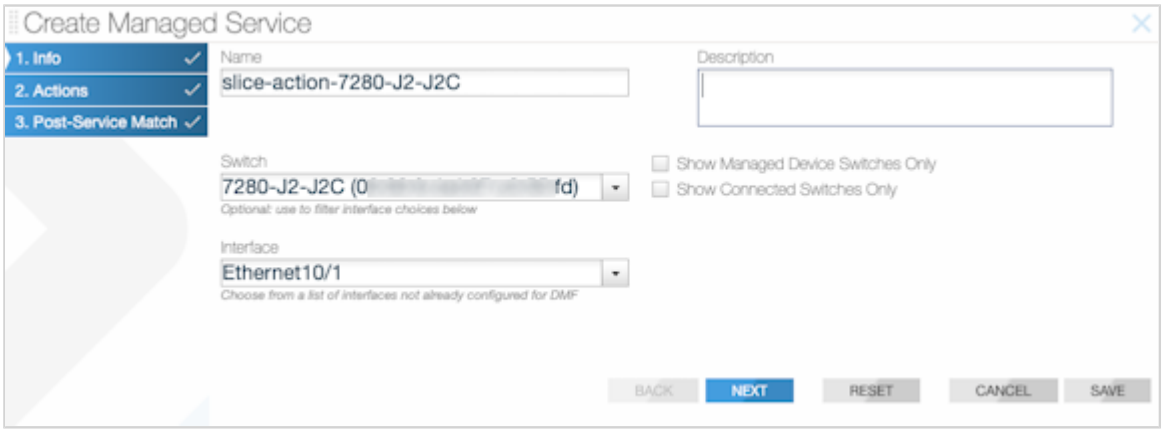


Figure 5-20: DMF Managed Services Add Managed Service



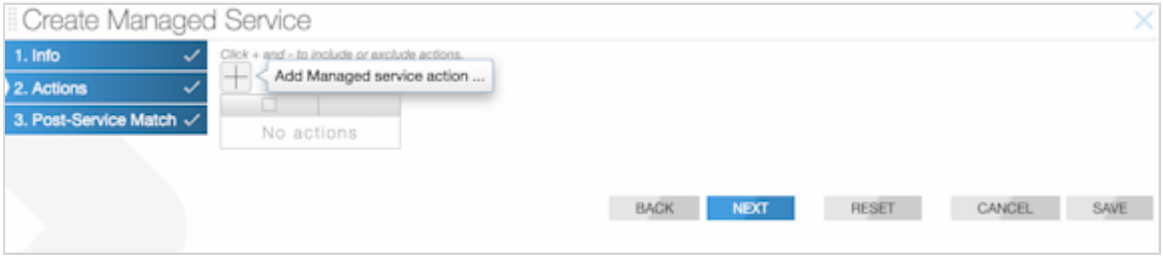
2. Configure a managed service interface on a switch that supports packet slicing. Make sure to deselect the **Show Managed Device Switches Only** checkbox.

Figure 5-21: Create Managed Service



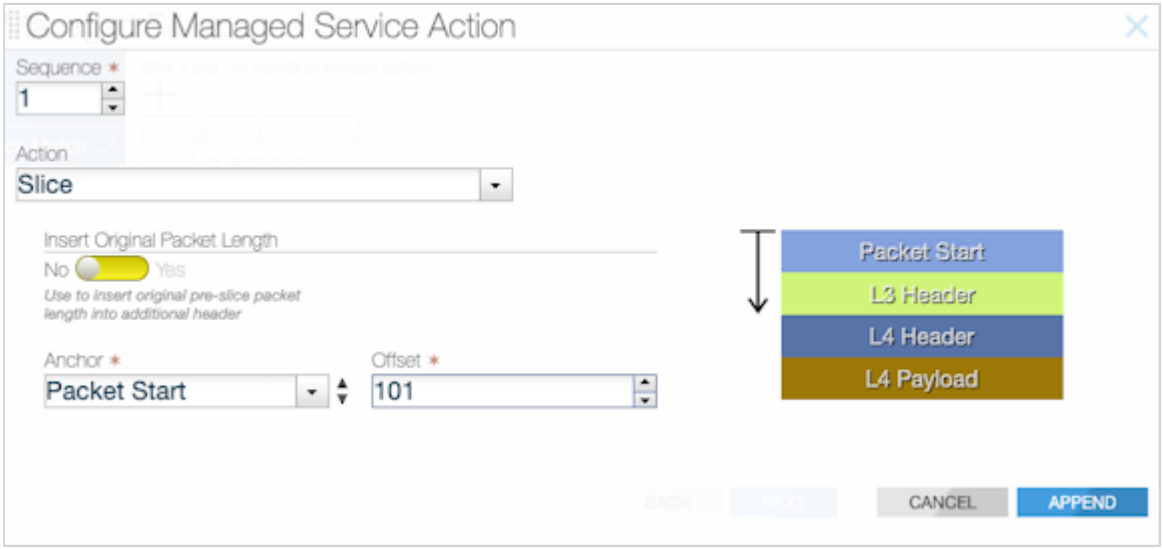
3. Configure a new managed service action using **Add Managed service action**. The action chain supports only one action when configuring packet slicing on a switch.

Figure 5-22: Add Managed service action



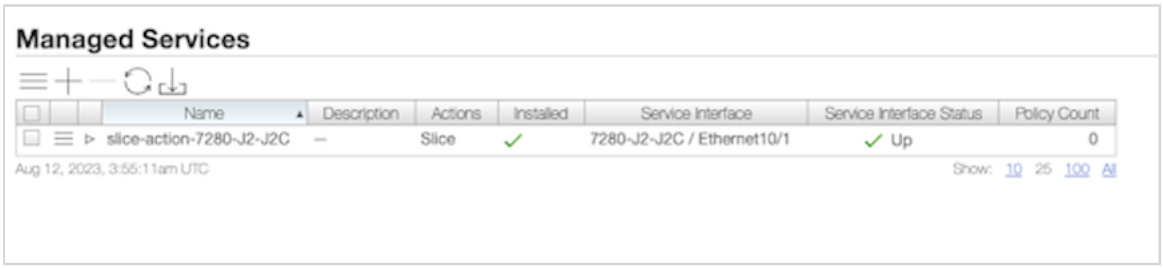
4. Use **Action** > **Slice** with **Anchor** > **Packet Start** to configure the packet slicing managed service on a switch.

Figure 5-23: Configure Managed Service Action



5. Select **Append** to continue. The slice action appears on the **Managed Services** page.

Figure 5-24: Slice Action Added

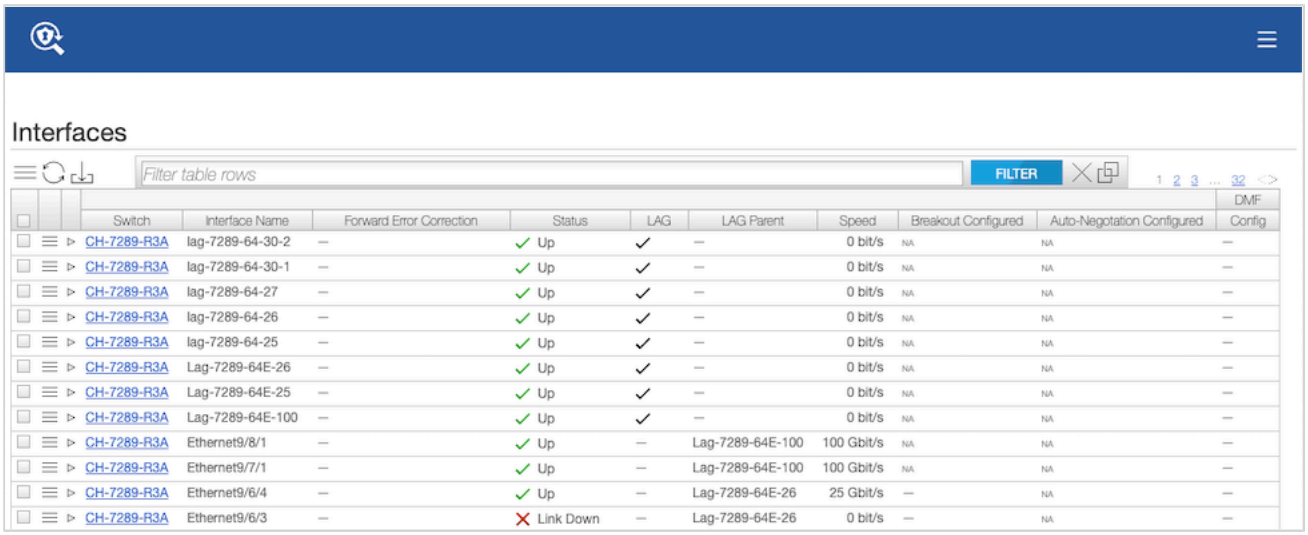


Interface Loopback Configuration

The managed service interface used for slice action must be in MAC loopback mode.

6. Configure the loopback mode in the **Fabric > Interfaces** page by selecting the **configuration icon** of the interface.

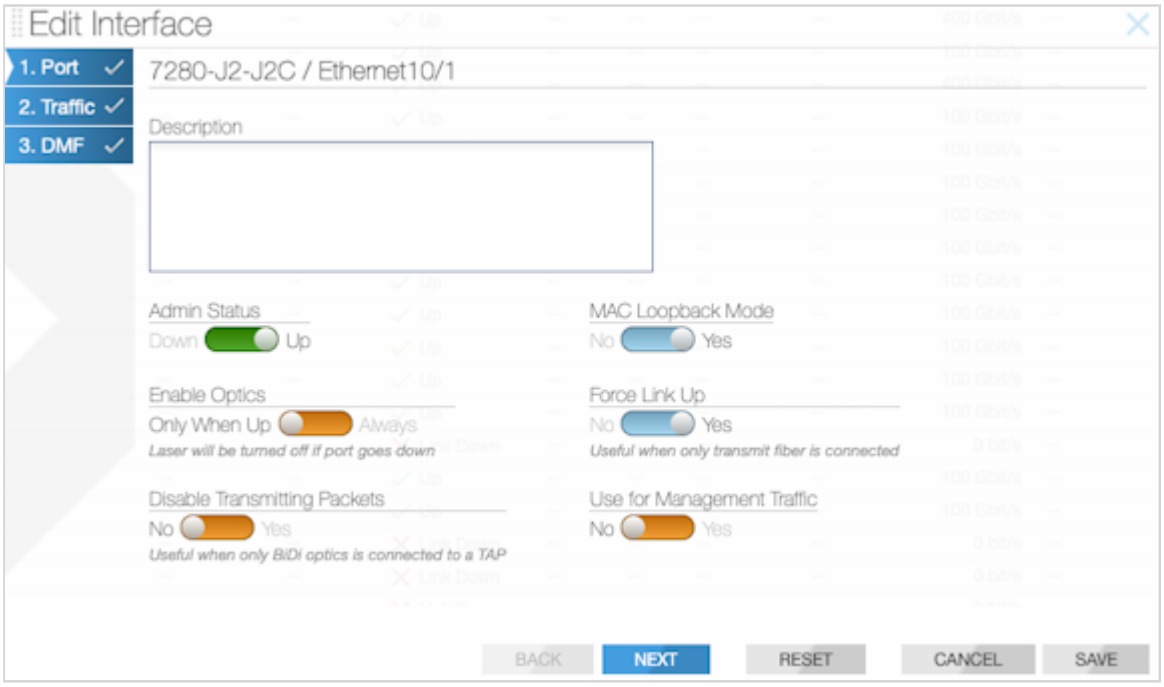
Figure 5-25: Interfaces



Note: The image above has been edited for documentation purposes. The actual output will differ.

7. Enable the toggle for **MAC Loopback Mode** (set the toggle to **Yes**).

Figure 5-26: Edit Interface

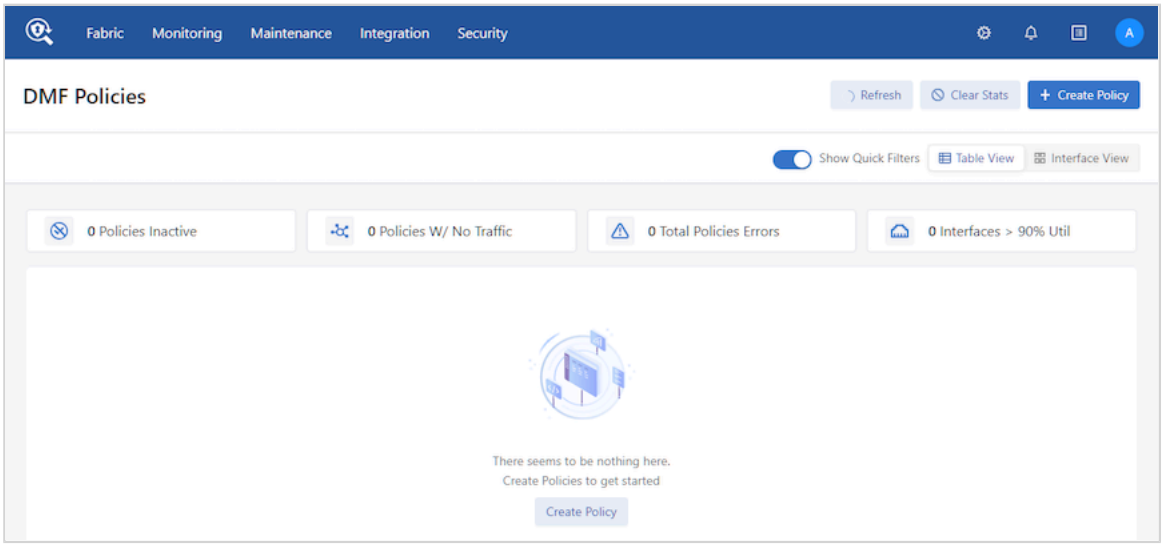


8. After all configuration changes are done **Save** the changes.

Policy Configuration

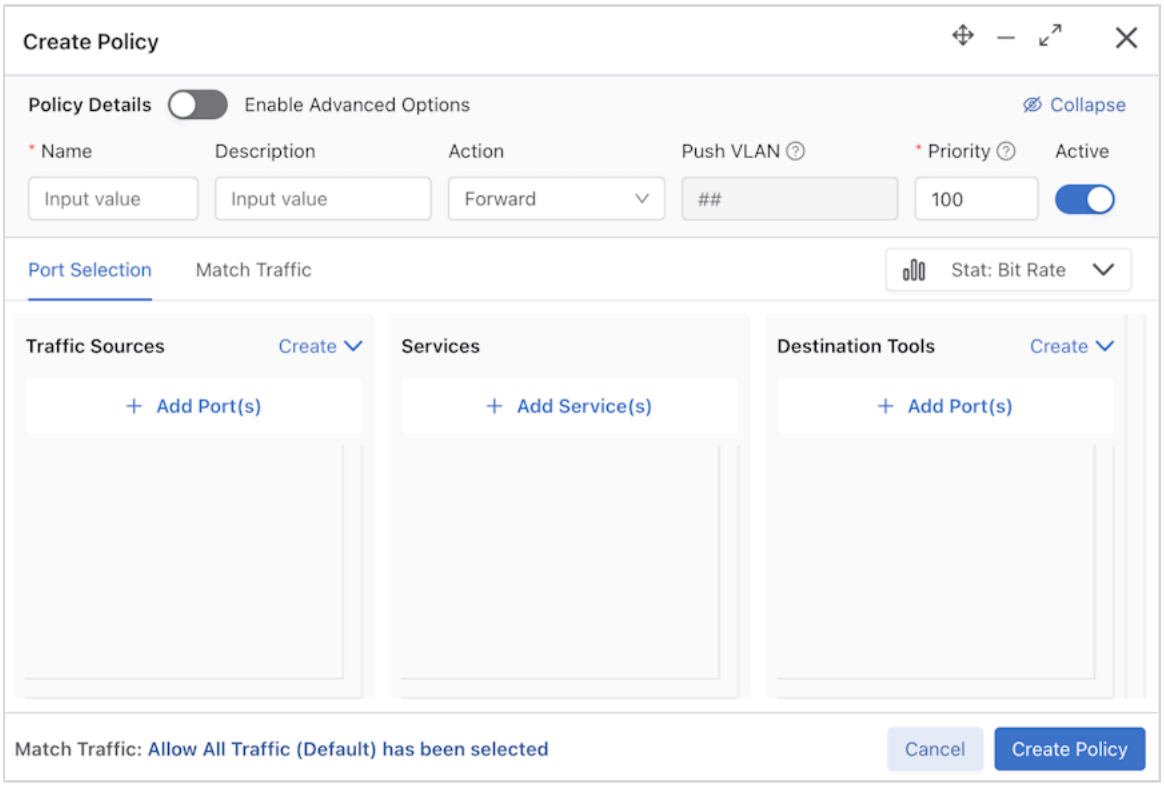
9. Create a new policy from the **DMF Policies** page.

Figure 5-27: DMF Policies Page



10. Add the previously configured packet slicing managed service.

Figure 5-28: Create Policy



11. Select **Add Service** under the **+ Add Service(s)** option shown above.

Figure 5-29: Add Service

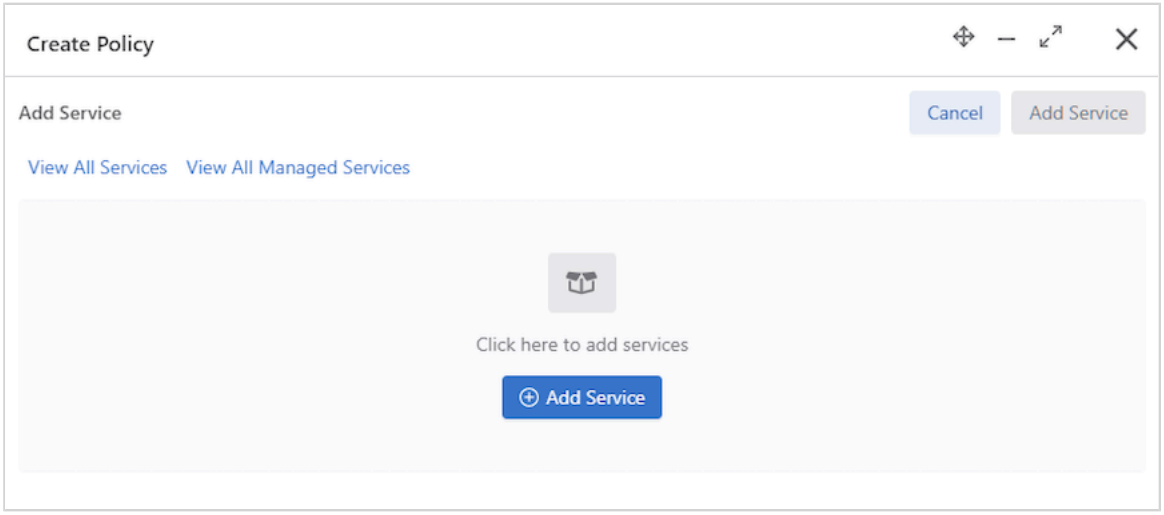
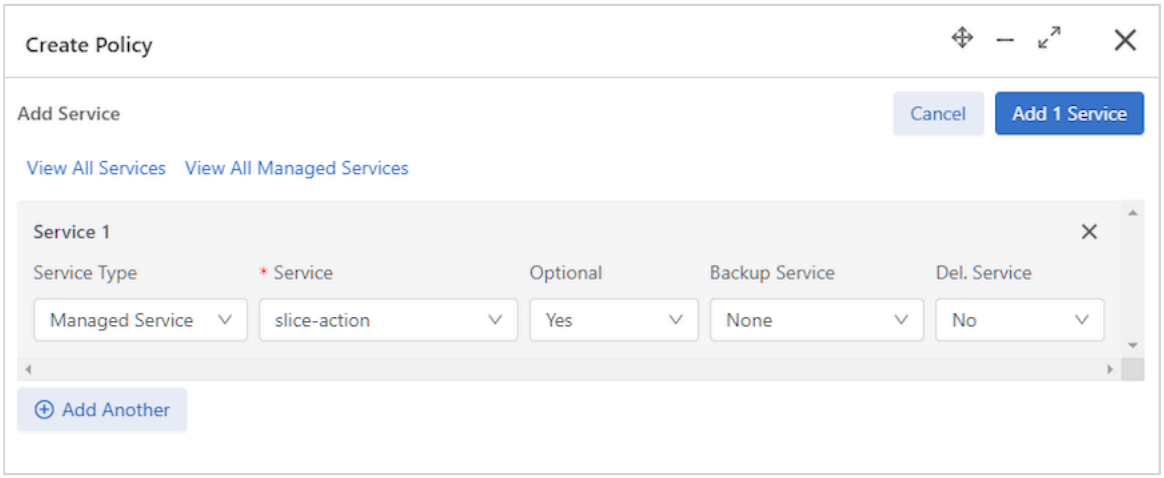
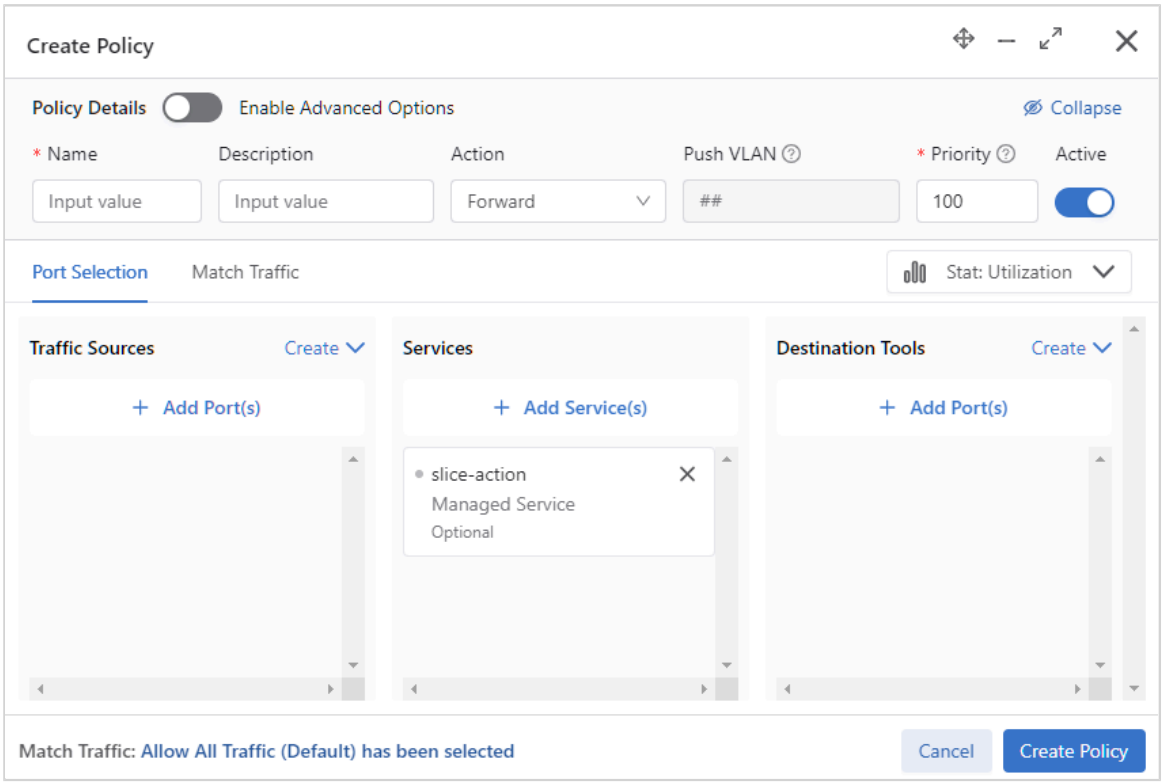


Figure 5-30: Service Type - Service - Slice Action



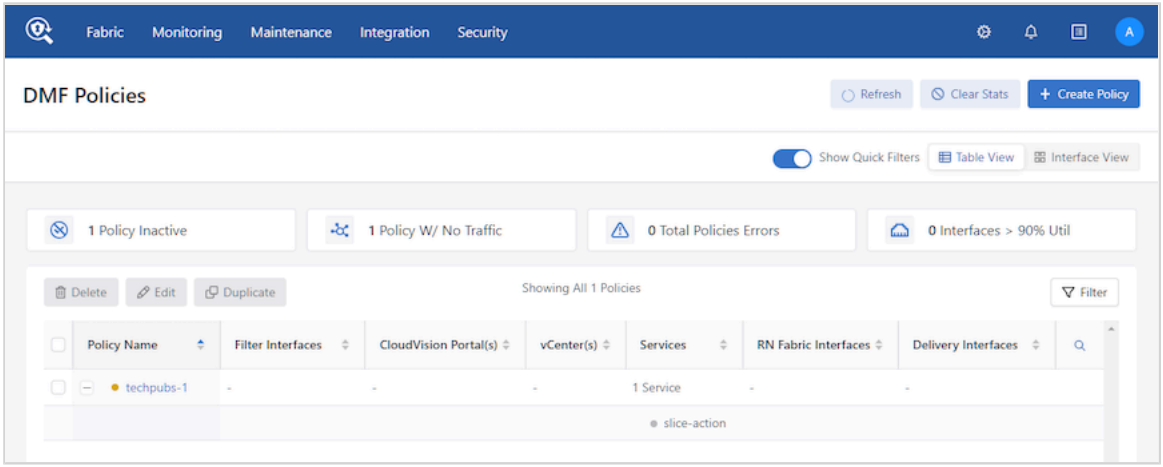
12. Select **Add 1 Service** and the slice-managed service (packet-slicing-policy) appears in the **Create Policy** page.

Figure 5-31: Manage Service Added



13. Select **Create Policy** and the new policy appears in **DMF Policies**.

Figure 5-32: DMF Policy Configured



Note: The images above have been edited for documentation purposes. The actual outputs may differ.

5.13.3 Troubleshooting Packet Slicing

The **show switch all property** command provides upper and lower bounds of packet slicing action's offset. If bounds are present, the feature is supported; otherwise, the switch does not support the packet slicing feature.

The **show fabric errors managed-service-error** command provides information when DANZ Monitoring Fabric (DMF) fails to install a configured packet slicing managed service on a switch.

The following are some of the failure cases:

1. The managed service interface is down.
2. More than one action is configured on a managed service interface of the switch.
3. The managed service interface on a switch is neither a physical interface nor a LAG port.
4. A non-slice managed service is configured on a managed service interface of a switch.
5. The switch does not support packet slicing managed service, and its interface is configured with slice action.
6. Slice action configured on a switch interface is not using a packet-start anchor.
7. The managed service interface is not in MAC loopback mode.

Use the following commands to troubleshoot packet-slicing issues.

```
Controller# show fabric errors managed-service-error
~~~
```

5.13.4 Packet Slicing Considerations

1. Managed service action chaining is not supported when using a switch interface as a managed service interface.

2. When configured for a supported switch, the managed service interface for slice action can only be a physical interface or a LAG.

3. When using packet slicing managed service, packets ingressing on the managed service interface are not counted in the ingress interface counters, affecting the output of the **show switch *switch name* interface *interface name* stats** and **show switch *switch name* interface *interface name* dmf-stats** commands. This issue does not impact byte counters; all byte counters will show the original packet size, not the truncated size.

4. A Dynamic Overlap policy causes all DMF-8.7.0 compatible 7820 switches to append an additional **Vlan Tag Post Slice Service** action. The default behavior of a fabric deployed in **Push-per-Policy** mode handles **Single Vlan Strip** at the delivery interface, removing the additional VLAN tag and leaving the original policy VLAN tag intact. In a non-default fabric deployment (**Push-per-filter**), to deliver the original packet to the tool node after a slice action, enable **strip-two-vlan** on the delivery interface.

5.14 VXLAN Stripping on the 7280R3 Switch

Virtual Extensible LAN Header Stripping

Virtual Extensible LAN (VXLAN) Header Stripping supports the delivery of decapsulated packets to tools and devices in a DANZ Monitoring Fabric (DMF) fabric. This feature removes the VXLAN header, previously established in a tunnel for reaching the TAP Aggregation switch or inherent to the tapped traffic within the DMF. Within the fabric, DMF supports the installation of the strip VXLAN service on a filter interface or a filter-and-delivery interface of a supported switch.

Platform Compatibility

For DMF deployments, the target platform is DCS-7280R3.

Use the **`show switch all property`** command to verify which switch in the DMF fabric supports this feature.

The feature is supported if the **`Strip Header Supported`** property has the value **`BSN_STRIP_HEADER_CAPS_VXLAN`**.



Note: The following example is displayed differently for documentation purposes than what appears when using the CLI.

```
# show switch all property
#                                   : 1
Switch                              : lyd599
Max Phys Port                       : 1000000
Min Lag Port                        : 1000001
Max Lag Port                        : 1000256
Min Tunnel Port                     : 15000001
Max Tunnel Port                     : 15001024
Max Lag Comps                       : 64
Tunnel Supported                    : BSN_TUNNEL_L2GRE
UDF Supported                       : BSN_UDF_6X2_BYTES
Enhanced Hash Supported             : BSN_ENHANCED_HASH_L2GRE,BSN_ENHANCED_HASH_L3,BSN_ENH
ANCED_HASH_L2,
                                      BSN_ENHANCED_HASH_MPLS,BSN_ENHANCED_HASH_SYMMETRIC
Strip Header Supported              : BSN_STRIP_HEADER_CAPS_VXLAN
Min Rate Limit                      : 1Mbps
Max Multicast Replication Groups    : 0
Max Multicast Replication Entries   : 0
PTP Timestamp Supported Capabilities : ptp-timestamp-cap-replace-smac, ptp-timestamp-cap-header-6
4bit,
                                      ptp-timestamp-cap-header-48bit, ptp-timestamp-cap-flow-bas
ed,
                                      ptp-timestamp-cap-add-header-after-l2
Min Truncate Offset                 : 100
Max Truncate Offset                 : 9236
```

5.14.1 Using the CLI to Configure VXLAN Header Stripping

Configuration

Use the following steps to configure **strip-vxlan** on a switch:

1. Set the optional field **strip-vxlan-udp-port** at switch configuration, and the default **udp-port** for **strip-vxlan** is **4789**.
2. Enable or disable **strip-vxlan** on a **filter** or **both-filter-and-delivery interface** using the **role both-filter-and-delivery interface-name *filter-interface* strip-vxlan** command.

```
> enable
# config
(config)# switch switch-name
(config-switch)# strip-vxlan-udp-port udp-port-number
(config-switch)# interface interface-name
(config-switch-if)# role both-filter-and-delivery interface-name filter-interface strip-vxlan
(config-switch-if)# role both-filter-and-delivery interface-name filter-interface no-strip-vxla
n
(config)# show running-config
```

3. After enabling a filter interface with **strip-vxlan**, any policy can use it. From the **config-policy** submode, add the **filter-interface** to the policy:

```
(config)# policy p1
(config-policy)# filter-interface filter-interface
```

Proceed to Show Commands.

Show Commands

Use the **show policy *policy name*** command to see the runtime state of a policy using a filter interface with **strip-vxlan** configured. It will also show the service interface information and stats.

```
# show policy strip-vxlan
Policy Name                             : strip-vxlan
Config Status                           : active - forward
Runtime Status                          : installed
Detailed Status                         : installed - installed to forward
Priority                                : 100
Overlap Priority                        : 0
# of switches with filter interfaces    : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces  : 0
# of filter interfaces                  : 1
# of delivery interfaces                : 1
# of core interfaces                    : 0
# of services                           : 0
# of pre service interfaces             : 0
# of post service interfaces            : 0
Push VLAN                               : 1
Post Match Filter Traffic               : -
Total Delivery Rate                     : -
Total Pre Service Rate                  : -
Total Post Service Rate                 : -
Overlapping Policies                    : none
Component Policies                      : none
Installed Time                          : 2024-05-02 19:54:27 UTC
Installed Duration                      : 1 minute, 18 secs
Timestamping enabled                    : False
~ Match Rules ~
# Rule
-|-----------|
1 1 match any
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~
# DMF IF Switch IF Name     State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time

-|------|------|-----------|-----|---|-------|-----|--------|--------|--------------------
----------|
1 f1     lyd598 Ethernet1/1 up    rx  0       0     0        -        2024-05-02 19:54:27.141000
 UTC
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~
# DMF IF Switch IF Name     State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time

-|------|------|-----------|-----|---|-------|-----|--------|--------|--------------------
----------|
1 d1     lyd598 Ethernet2/1 up    tx  0       0     0        -        2024-05-02 19:54:27.141000
 UTC
~ Service Interface(s) ~
None.
~ Core Interface(s) ~
None.
~ Failed Path(s) ~
None.
```

5.14.2 Using the GUI to Configure VXLAN Header Stripping

Filter Interface Configuration

To configure or edit a filter interface, proceed to **Interfaces** from the **Monitoring** menu and select **Interfaces** > **Filter Interfaces**.

Figure 5-33: Filter Interfaces

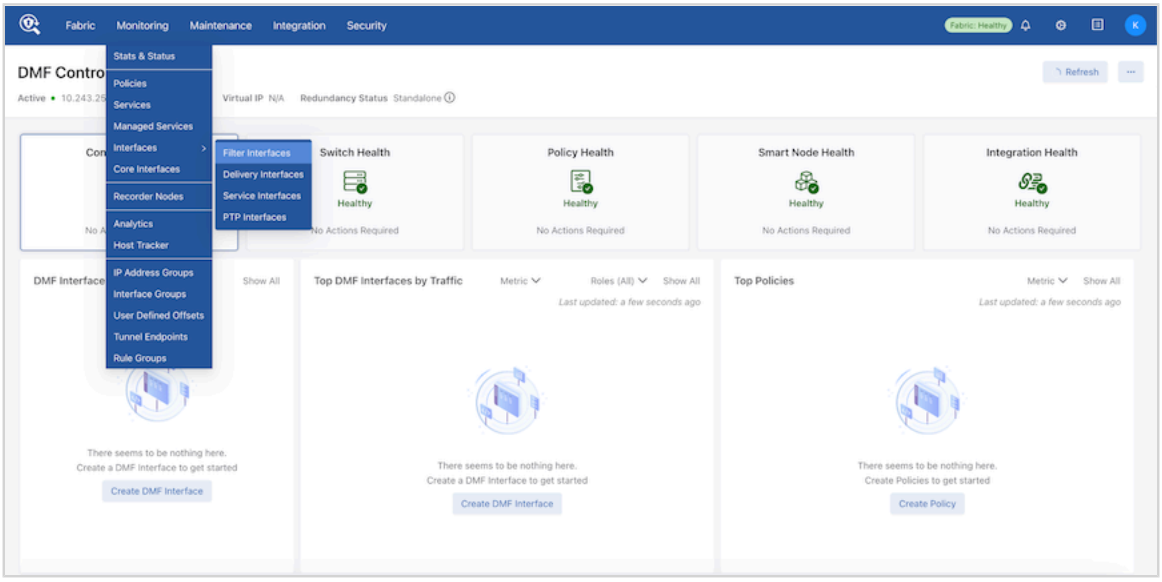
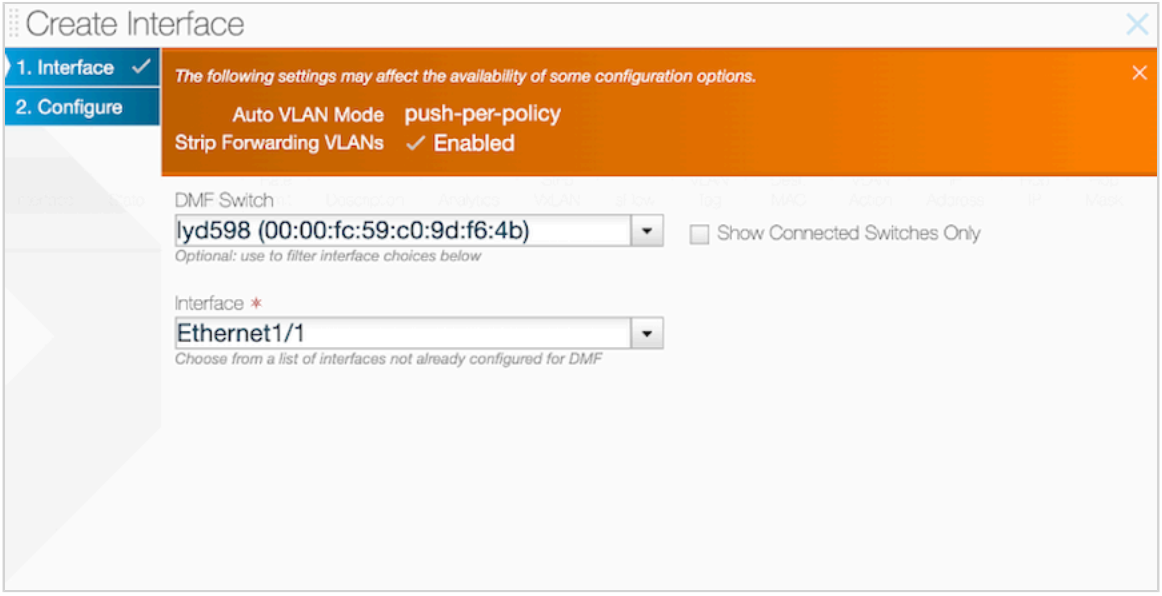


Figure 5-34: DMF Interfaces



Configure a filter interface on a switch that supports **`strip-vxlan`**.

Figure 5-35: Configure Filter Interface



Enable or **Disable** Strip VXLAN.

Figure 5-36: Enable Strip VXLAN

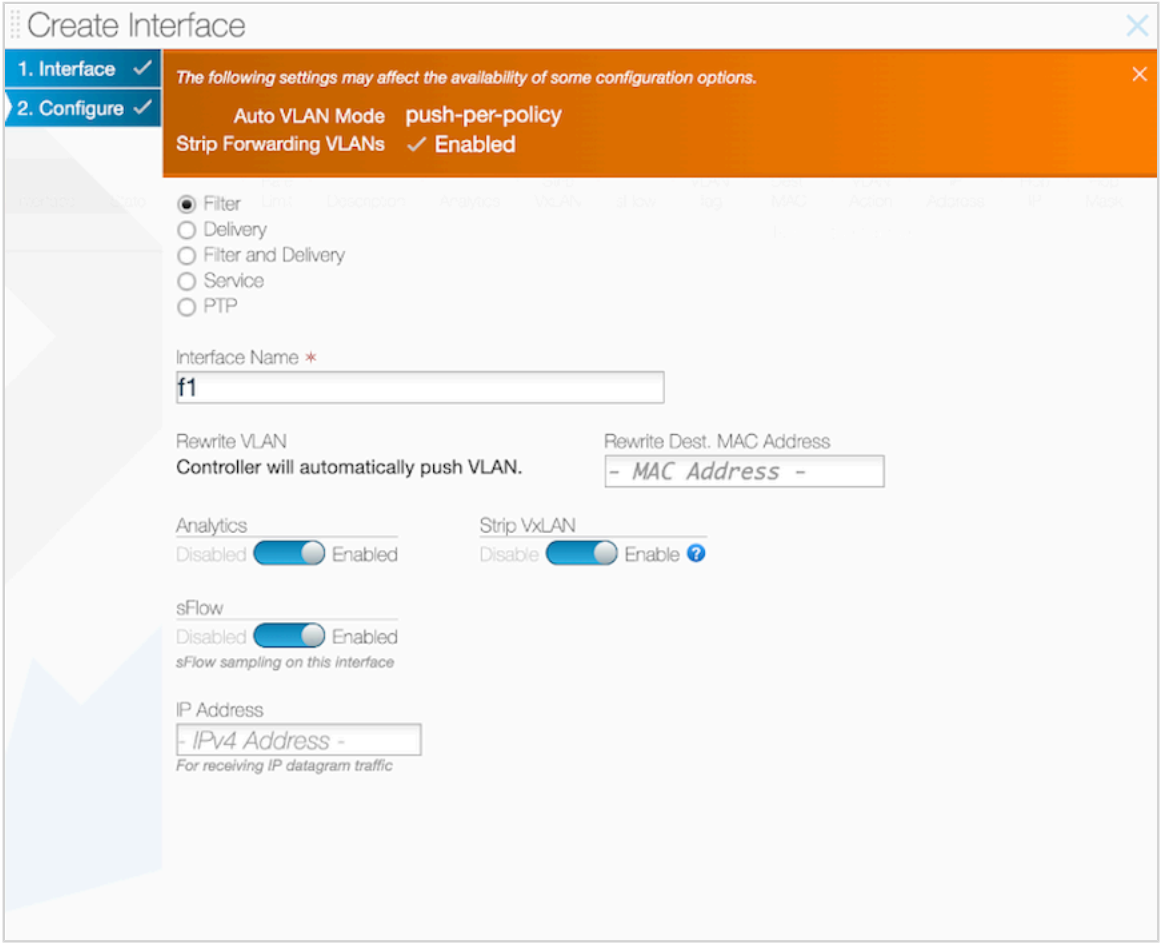
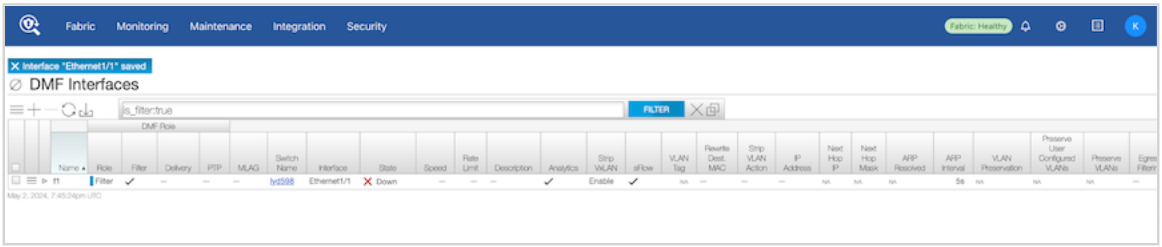


Figure 5-37: DMF Interfaces Updated



Policy Configuration

Create a new policy using **DMF Policies** and add the filter interface with strip VXLAN enabled.

Figure 5-38: Create Policy

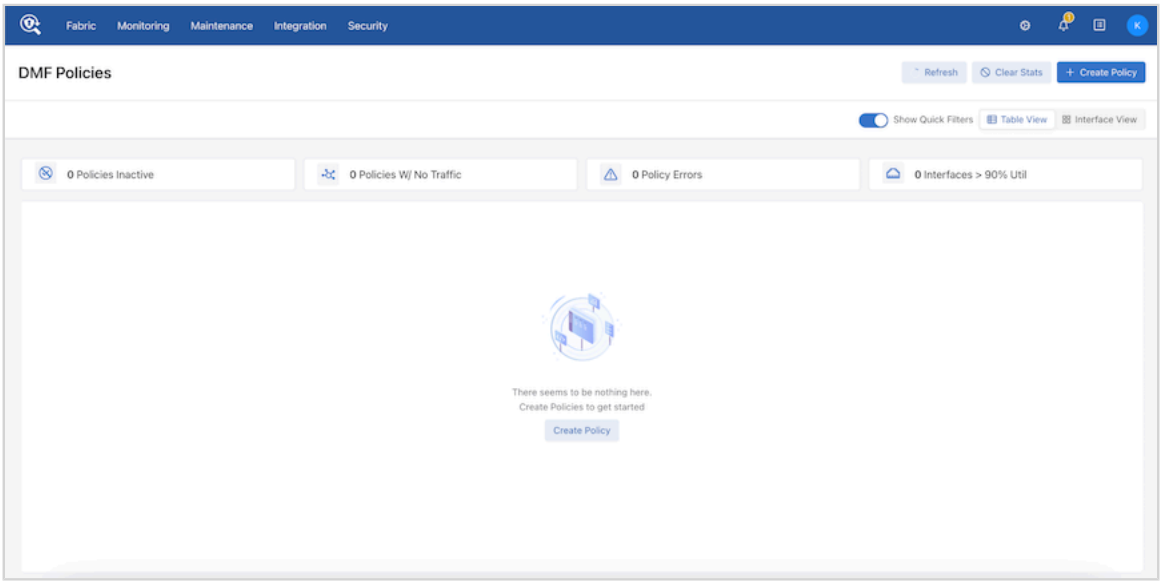
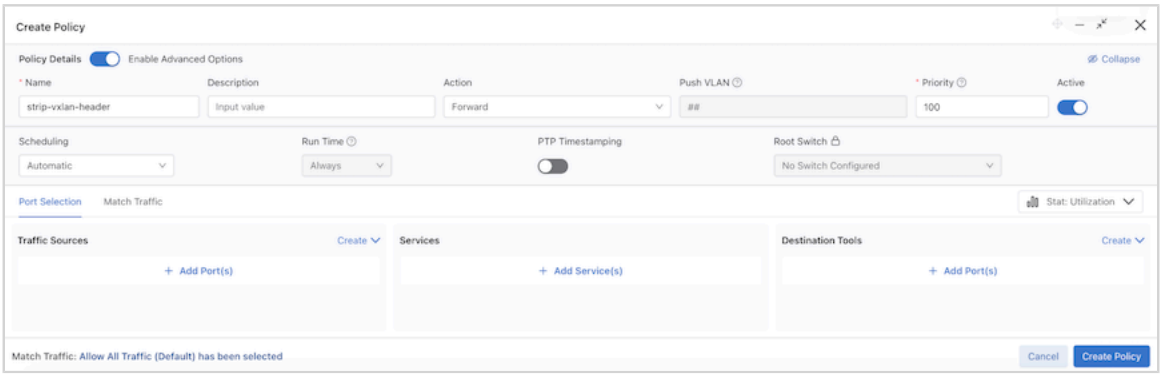
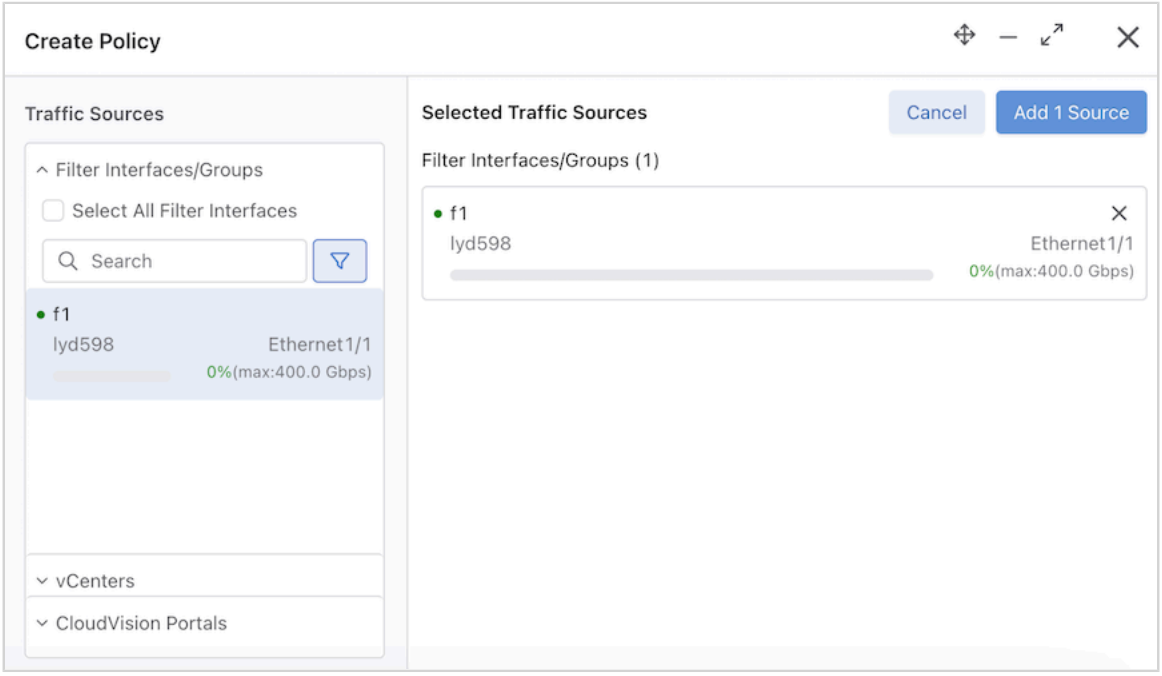


Figure 5-39: Strip VXLAN Header



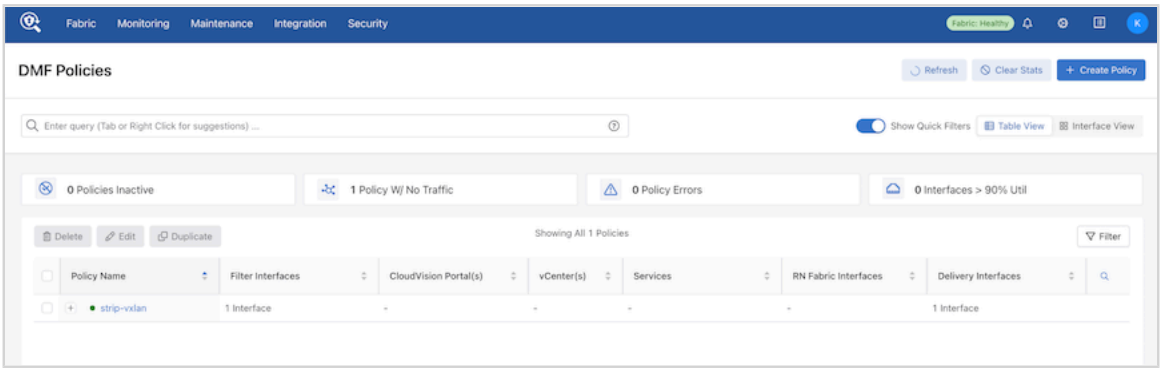
Select **Add port(s)** under the **Traffic Sources** option and add the **Filter Interface**.

Figure 5-40: Selected Traffic Sources



Add another delivery interface and create the policy.

Figure 5-41: Policy Created



5.14.3 Syslog Messages

There are no syslog messages associated with this feature.

5.14.4 Troubleshooting

The **show switch all property** command provides the **Strip Header Supported** property of the switch. If the value **BSN_STRIP_HEADER_CAPS_VXLAN** is present, the feature is supported; otherwise, the switch does not support this feature.

The **show fabric warnings feature-unsupported-on-device** command provides information when DMF fails to enable **strip-vxlan** on an unsupported switch.

The **show switch *switch-name* table *strip-vxlan-header*** command provides the gentable details.

The following are examples of several failure cases:

1. The filter interface is down.
2. The interface with **strip-vxlan** is neither a filter interface nor a filter-and-delivery interface.
3. The switch does not support **strip-vxlan**.
4. Tunneling / UDF is enabled simultaneously with **strip-vxlan**.
5. Unsupported pipeline mode with **strip-vxlan** enabled (**strip-vxlan** requires a specific pipeline mode **strip-vxlan-match-push-vlan**).

5.14.5 Limitations

- When configured for a supported switch, the filter interface for **decap-vxlan** action can only be a physical interface or a LAG.
- It is not possible to enable **strip-vxlan** simultaneously with tunneling / UDF.
- When enabling **strip-vxlan** on one or more switch interfaces on the same switch, other filter interfaces on the same switch cannot be matched on the VXLAN header.

5.15 Session Slicing for TCP and UDP Sessions

Session-slice keeps track of TCP and UDP sessions (distinguished by source and destination IP address and port) and counts the number of packets sent in each direction (client-to-server and vice versa). After recognizing the session, the action transmits a user-configured number of packets to the tool node.

For TCP packets, session-slice tracks the number of packets sent in each direction after establishing the TCP handshake. Slicing begins after the packet count in a direction has reached the configured threshold in both directions.

For UDP packets, slicing begins after reaching the configured threshold in either direction.

By default, session-slice will operate on both TCP and UDP sessions but is configurable to operate on only one or the other.



Note: The count of packets in one direction may exceed the user-configured threshold because fewer packets have arrived in the other direction. Counts in both directions must be greater than or equal to the threshold before dropping packets.

Refer to the *DANZ Monitoring Fabric (DMF) Verified Scale Guide* for session-slicing performance numbers.

Configure **`session-slice`** in managed services through the Controller as a Service Node action.

- Using the CLI to Configure Session Slicing
- Using the GUI to Configure Session Slicing

5.15.1 Using the CLI to Configure Session Slicing

Configure **session-slice** in managed services through the Controller as a Service Node action.

Configuration Steps

1. Create a managed service and enter the service interface.
2. Choose the session-slice service action with the command: ***seq num* session-slice**

> **Note:** The ***seq num* session-slice** command opens the **session-slice submode**, which supports two configuration parameters: **slice-after** and **idle-timeout**.

3. Use **slice-after** to configure the packet threshold, after which the Service Node will stop forwarding packets to tool nodes.
4. Use **idle-timeout** to configure the timeout in milliseconds before an idle connection is removed from the cache. **idle-timeout** is an optional command with a default value of **60000** ms.

```
dmf-controller-1(config)# managed-service managed_service_1
dmf-controller-1(config-managed-srv)# 1 session-slice
dmf-controller-1(config-managed-srv-ssn-slice)# slice-after 1000
dmf-controller-1(config-managed-srv-ssn-slice)# idle-timeout 60000
```

Show Commands

The following **show** commands provide helpful information.

The **show running-config managed-service *managed service*** command helps verify whether the **session-slice** configuration is complete.

```
dmf-controller-1(config)# show running-config managed-service managed_service_1

! managed-service
managed-service managed_service_1
!
1 session-slice
slice-after 1000
idle-timeout 60000
```

The **show managed-services *managed service*** command provides status information about the service.

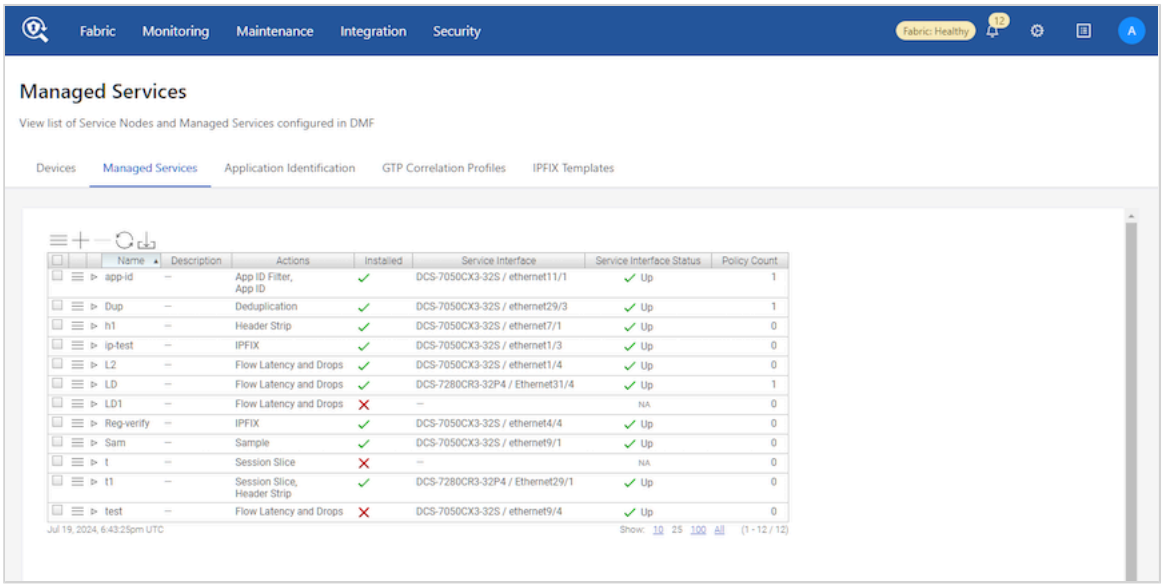
```
dmf-controller-1(config)# show managed-services managed_service_1
# Service Name      Switch          Switch Interface Installed Max Post-Service BW Max Pre-Service
 BW Total Post-Service BW Total Pre-Service BW
-|-----------------|---------------|----------------|---------|-------------------|-------
-----------|---------------------|--------------------|
1 managed_service_1 DCS-7050CX3-32S ethernet2/4      True      25Gbps              25Gbps
    624Kbps               432Mbps
```

5.15.2 Using the GUI to Configure Session Slicing

Perform the following steps to configure session slicing.

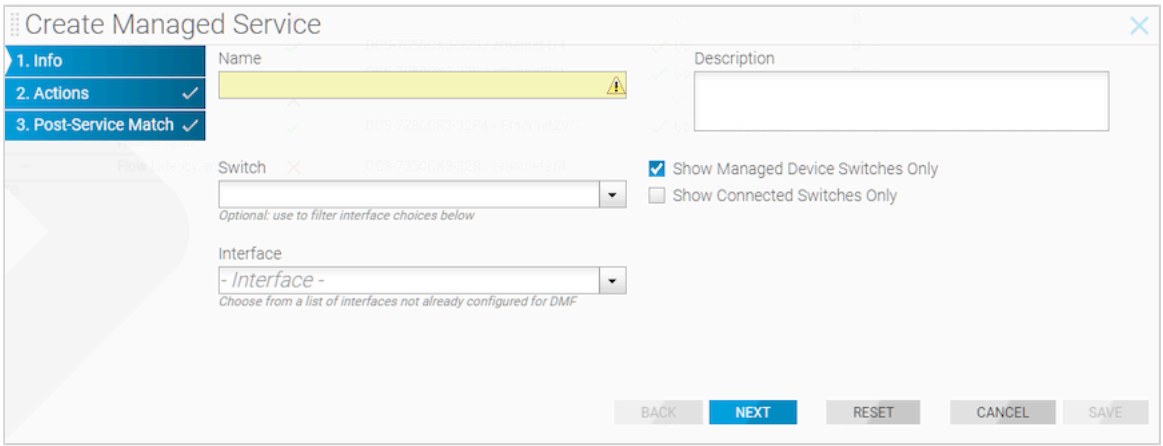
1. Navigate to **Monitoring** > **Managed Services** > **Managed Services**.

Figure 5-42: Managed Services



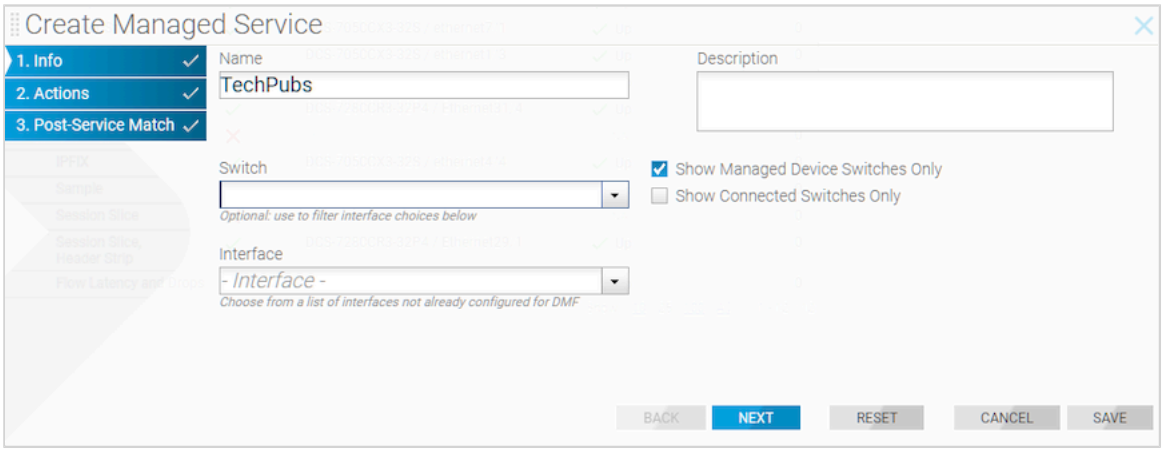
2. Select the **+** icon to create a new managed service.

Figure 5-43: Create Managed Service



3. Enter a **Name** for the managed service.

Figure 5-44: Managed Service Name



4. Select a **Switch** from the drop-down list.

Figure 5-45: Manage Service Switch

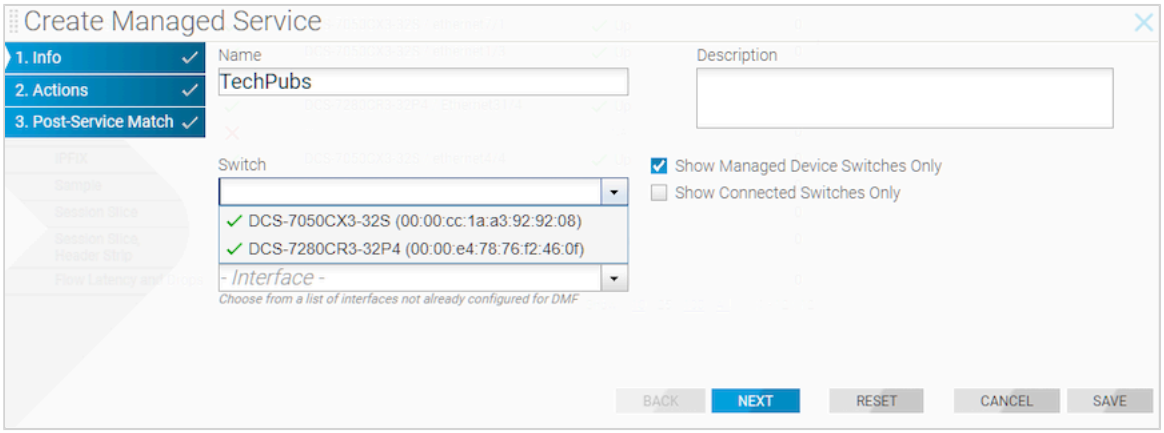
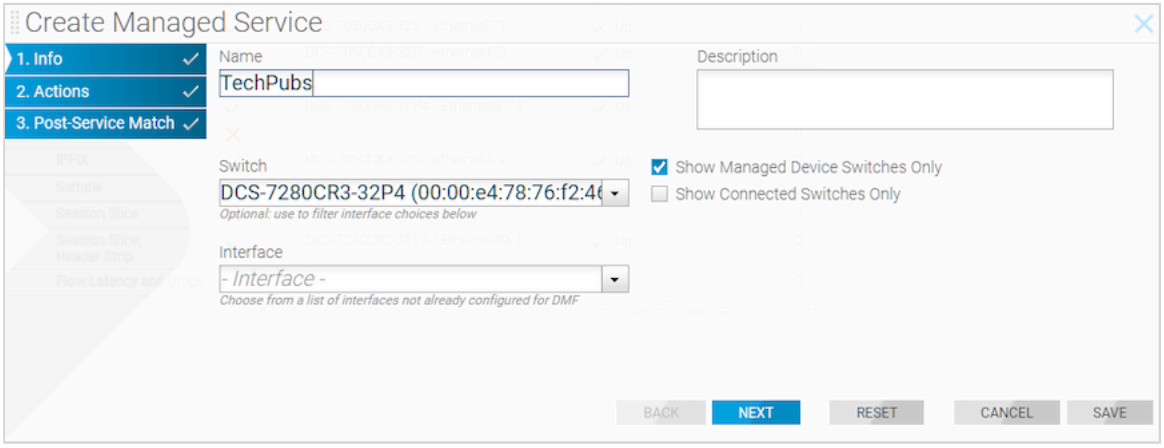
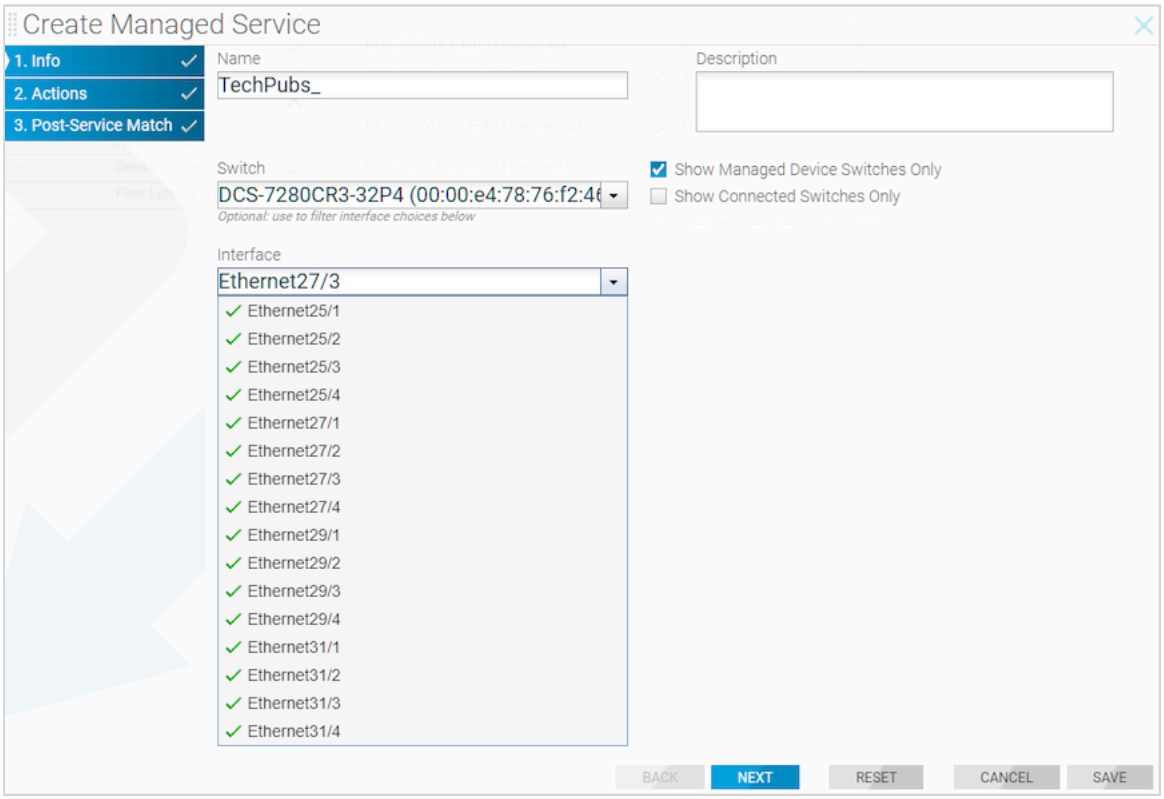


Figure 5-46: Managed Service Switch Added



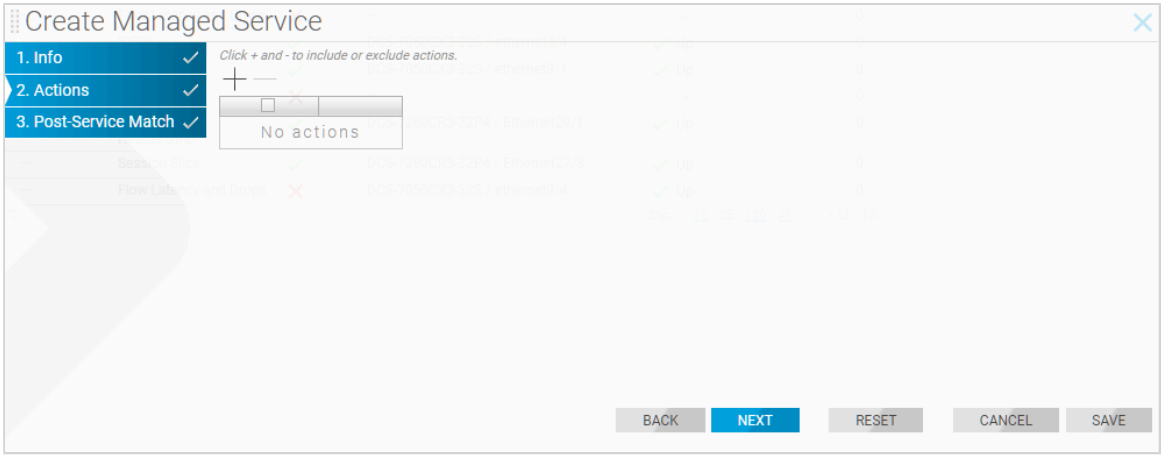
5. Select an **Interface** from the drop-down list.

Figure 5-47: Managed Service Interface Added



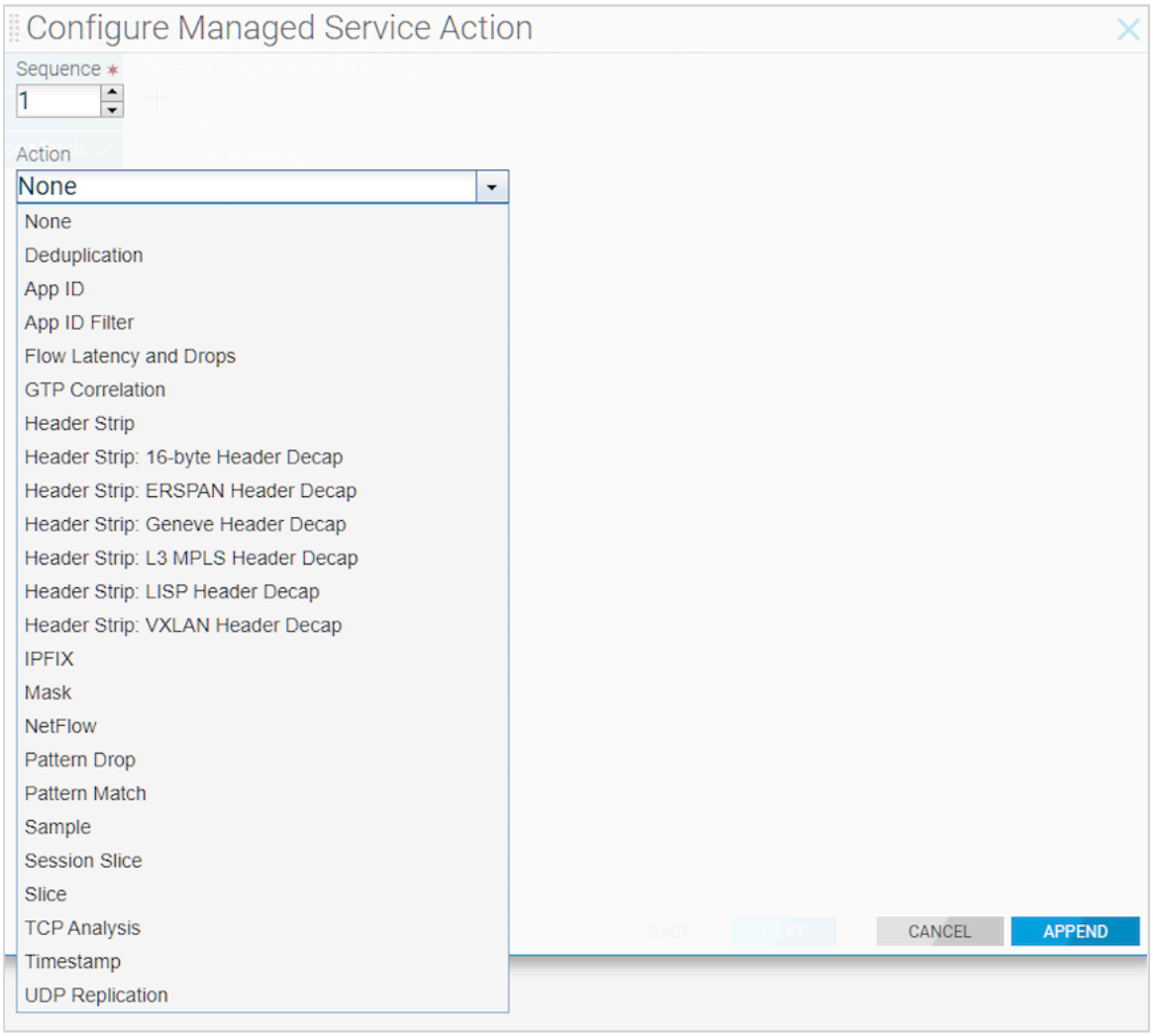
6. Select **Actions** or **Next**.

Figure 5-48: Actions Menu



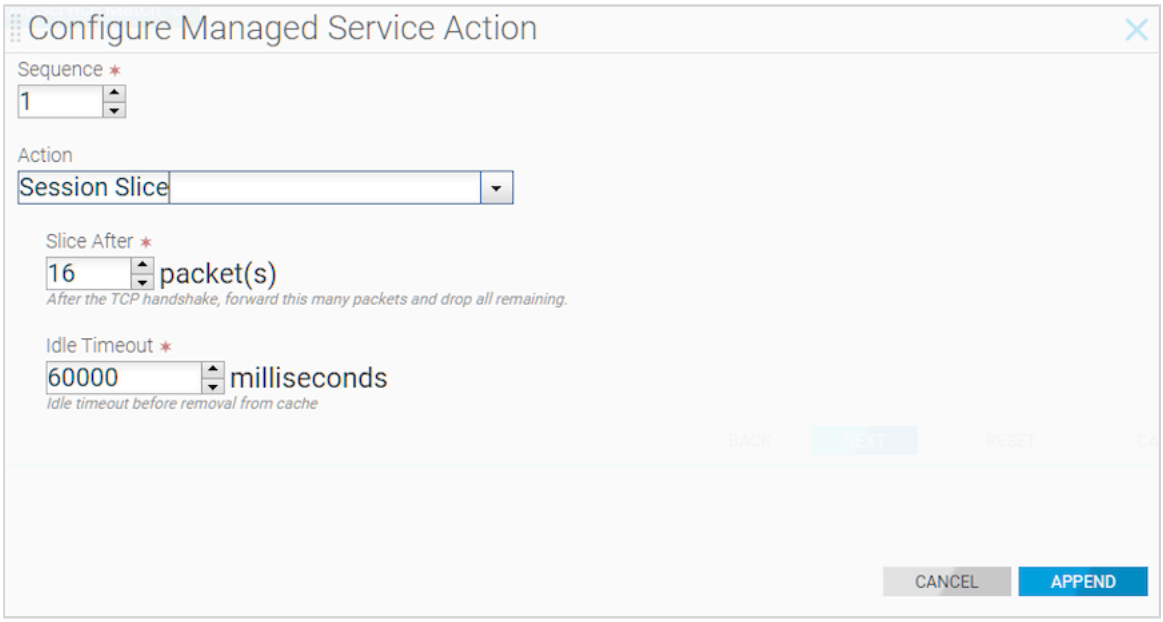
7. Select the **+** icon to select a managed service action.

Figure 5-49: Configure Managed Service Action List



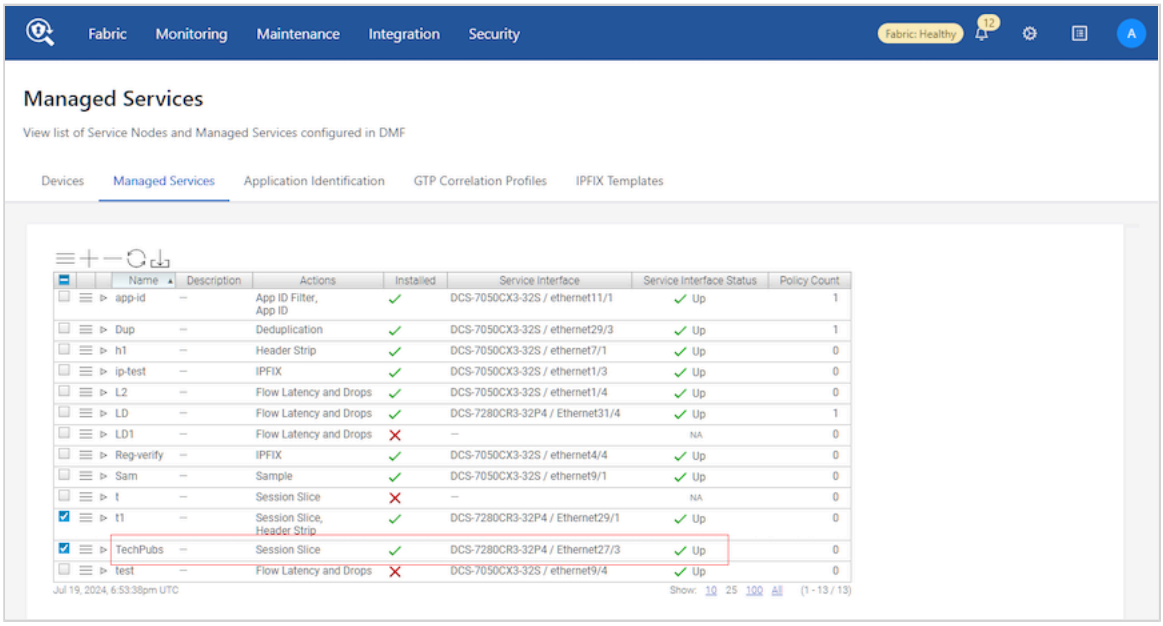
8. Choose **Session Slice** from the drop-down list. Adjust the **Slice After** and **Idle Timeout** parameters, as required.

Figure 5-50: Configure Managed Service Action Session Slice



9. Select **Append** and then **Save** to add the session slice managed service.

Figure 5-51: Managed Service Session Slice

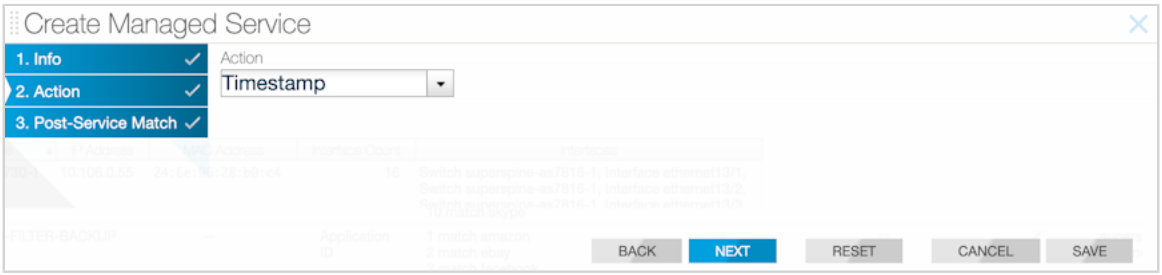


5.16 Timestamp Action

The timestamp service action identifies and timestamps every packet it receives with the time the service node receives the packet for matching traffic.

GUI Configuration

Figure 5-52: Create Managed Service: Timestamp Action



CLI Configuration

```
! managed-service
managed-service MS-TIMESTAMP-1
1 timestamp
service-interface switch CORE-SWITCH-1 ethernet15/3
```

5.17 UDP-Replication Action

The UDP-replication service action copies UDP messages, such as Syslog or NetFlow messages, and sends the copied packets to a new destination IP address.

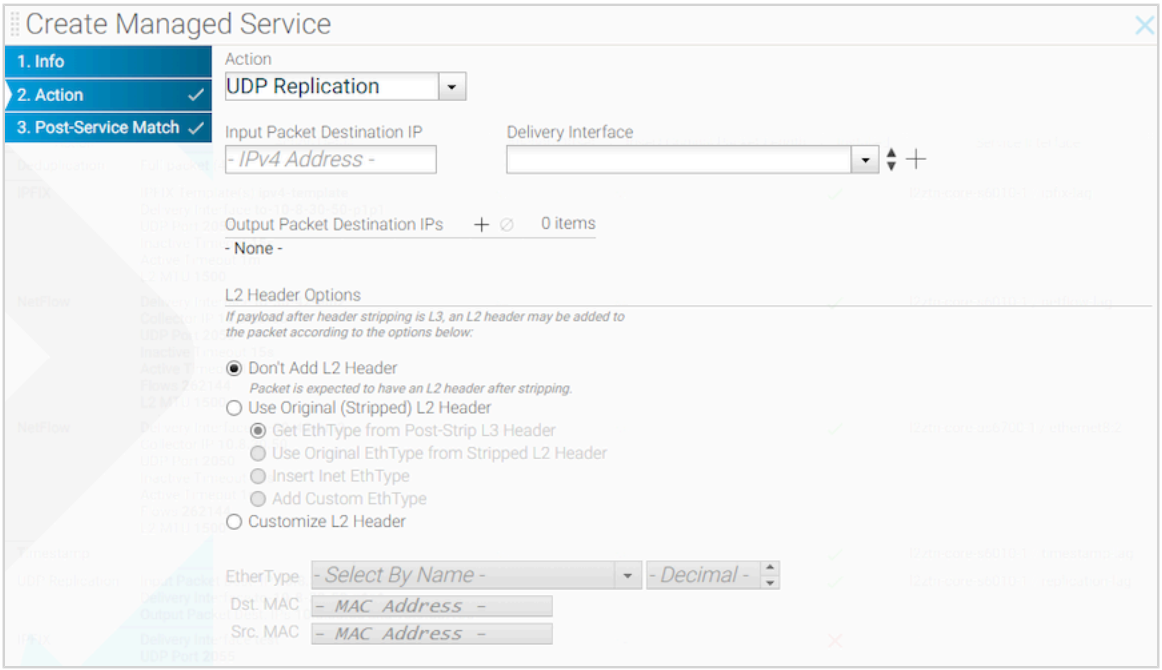
Configure a rate limit when enabling UDP replication. When upgrading from a version of DANZ Monitoring Fabric (DMF) before release ***6.3.1***, the UDP-replication configuration is not applied until a rate limit is applied to the delivery interface.

The following example illustrates applying a rate limit to a delivery interface used for UDP replication:

```
CONTROLLER-1(config)# switch DMF-DELIVERY-SWITCH-1
CONTROLLER-1(config-switch)# interface ethernet1
CONTROLLER-1(config-switch-if)# role delivery interface-name udp-delivery-1
CONTROLLER-1(config-switch-if)# rate-limit 256000
```

Note: No other service action can be applied after a UDP-replication service action.

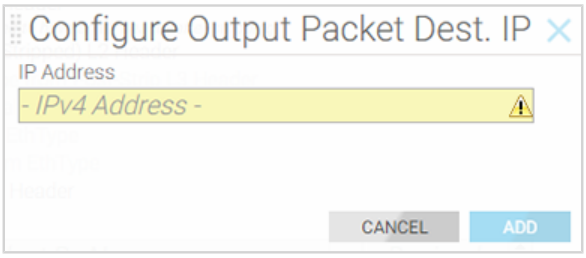
GUI Configuration



Use the UDP-replication service to copy UDP traffic, such as Syslog messages or NetFlow packets, and send the copied packets to a new destination IP address. This function sends traffic to more destination syslog servers or NetFlow collectors than would otherwise be allowed.

Enable the checkbox for the destination for the copied output, or select the provision control (**+**) and add the IP address in the dialog that appears.

Figure 5-53: Configure Output Packet Destination IP



For the header-strip service action only, configure the policy rules for matching traffic after applying the header-strip service action. After completing pages 1-4, select **Append** and enable the checkbox to apply the policy.

Select **Save** to save the managed service.

CLI Configuration

Enter the **`1 udp-replicate`** command and identify the configuration name (the submode changes to the ***config-managed-srv-udp-replicate*** submode) to view and configure a specific UDP-replication configuration.

```
controller-1(config)# managed-service MS-UDP-REPLICATE-1
controller-1(config-managed-srv)# 1 udp-replicate DELIVERY-INTF-TO-COLLECTOR
controller-1(config-managed-srv-udp-replicate)#
```

From this submode, define the destination address of the packets to copy and the destination address for sending the copied packets.

```
controller-1(config-managed-srv-udp-replicate)# in-dst-ip 10.1.1.1
controller-1(config-managed-srv-udp-replicate)# out-dst-ip 10.1.2.1
```

5.18 Redundancy of Managed Services in Same DMF Policy

In this method, users can use a second managed service as a backup service in the same DANZ Monitoring Fabric (DMF) policy. The backup service is activated only when the primary service becomes unavailable. The backup service can be on the same service node or core switch or a different service node and core switch.



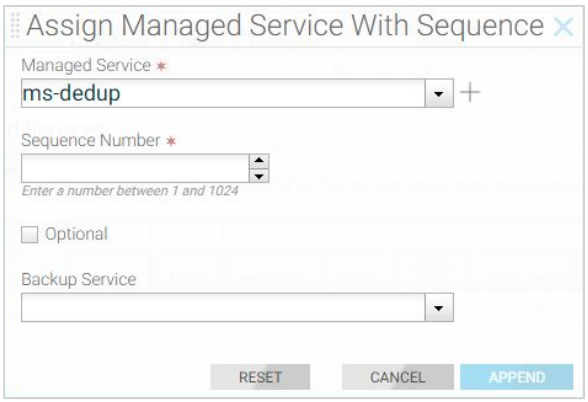
Note: Transitioning from active to backup managed service requires reprogramming switches and associated managed appliances. This reprogramming, done seamlessly, will result in a slight traffic loss.

5.18.1 Using the GUI to Configure a Backup Managed Service

To assign a managed service as a backup service in a DANZ Monitoring Fabric (DMF) policy, perform the following steps:

1. Select **Monitoring > Policies** and select the Provision control (**+**) to create a new policy.
2. Configure the policy as required. From the **Services** section, select the Provision control (**+**) in the **Managed Services** table.

   Figure 5-54: Policy with Backup Managed Service



3. Select the primary managed service from the **Managed Service** selection list.
4. Select the backup service from the **Backup Service** selection list and select **Append**.

5.18.2 Using the CLI to Configure a Backup Managed Service

To implement backup-managed services, perform the following steps:

1. Identify the first managed service.

```
managed-service MS-SLICE-1
1 slice l3-header-start 20
service-interface switch CORE-SWITCH-1 lag1
```

2. Identify the second managed service.

```
managed-service MS-SLICE-2
1 slice l3-header-start 20
service-interface switch CORE-SWITCH-1 lag2
```

3. Configure the policy referring to the backup managed service.

```
policy SLICE-PACKETS
action forward
delivery-interface TOOL-PORT-1
filter-interface TAP-PORT-1
use-managed-service MS-SLICE-1 sequence 1 backup-managed-service MS-SLICE-2
1 match ip
```

5.19 Application Identification

The DANZ Monitoring Fabric (DMF) Application Identification feature allows for the monitoring of applications identified with Deep Packet Inspection (DPI) into packet flows received via filter interfaces and generates IPFIX flow records. These IPFIX flow records are transmitted to a configured collector device via the L3 delivery interface. The feature provides a filtering function by forwarding or dropping packets from specific applications before sending the packet to the analysis tools.



Note: Application identification is supported on Service Nodes (DCA-DM-SC and DCA-DM-SC2) and Service Nodes (DCA-DM-SDL and DCA-DM-SEL).

5.19.1 Using the CLI to Configure Application Identification

Configure the feature through the Controller in managed services.

There are two application identification services to configure:

- app-id
- app-id-filter
- app-id and app-id-filter combined

5.19.2 Using the CLI to Configure app-id

Perform the following steps to configure app-id.

1. Create a managed service and enter the service interface.
2. Choose the app-id managed service using the ***seq num*** **app-id** command.

> **Note:** The above command should enter the app-id submode, which supports two configuration parameters: **collector** and **l3-delivery-interface**. Both are required.

3. To configure the IPFIX collector IP address, enter the following command: **collector** ***ip-address***. The UDP port and MTU parameters are optional; the default values are ***4739*** and ***1500***, respectively.
4. Enter the command: **l3-delivery-interface** ***delivery interface name*** to configure the delivery interface.
5. Add this managed service to a policy. The policy will not have a physical delivery interface.

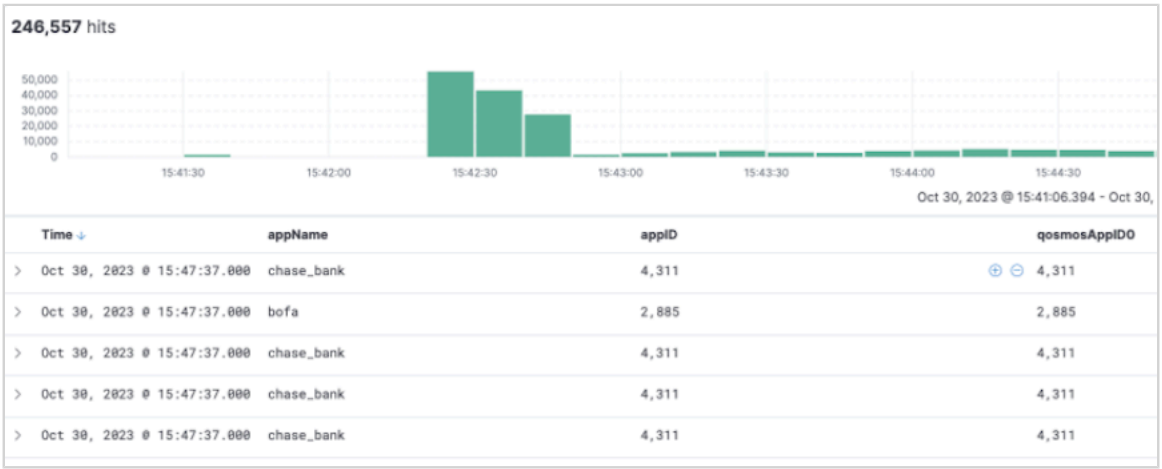
The following shows an example of an app-id configuration that sends IPFIX application records to the Collector (analytics node) at IP address 192.168.1.1 over the configured delivery interface named ***app-to-analytics***:

```
managed-service ms
service-interface switch core1 ethernet2
!
1 app-id
collector 192.168.1.1
l3-delivery-interface app-to-analytics
```

After configuring the app-id, refer to the analytics node for application reports and visualizations. For instance, a flow is classified internally with the following tuple: ***ip***, ***tcp***, ***http***, ***google***, and ***google_maps***. Consequently, the analytics node displays the most specific app ID for this flow as **google_maps** under **appName**.

On the Analytics Node, there are AppIDs 0-4 representing applications according to their numerical IDs. **0** is the most specific application identified in that flow, while **4** is the least. In the example above, **ID 0** would be the numerical ID for **google_maps**, **ID 1 google**, **ID 2 http**, **ID 3 tcp**, and **ID 4 IP address**. Use the **appName** in place of these since these require an ID to name mapping to interpret.

Figure 5-55:



5.19.3 Using the CLI to Configure app-id-filter

Perform the following steps to configure app-id-filter:

1. Create a managed service and enter the service interface.
2. Choose the app-id managed service using the ***seq num*** **app-id-filter** command.



Note: The above command should enter the **app-id-filter submode**, which supports three configuration parameters: **app**, **app-category**, and **filter-mode**. The category **app** is required, while **app-category** and **filter-mode** are optional. The option **filter-mode** has a default value of **forward**.

3. Enter the command: **app** ***application name*** to configure the application name.



Tip: Press the **Tab** key after entering the app keyword to see all possible application names. Type in a partial name and press the **Tab** to see all possible choices to auto-complete the name. The application name provided must match a name in this list of app names. A service node must be connected to the Controller for this list to appear. Any number of apps can be entered one at a time using the **app application-name** command. An example of a (partial) list of names:

```
dmf-controller-1 (config-managed-srv-app-id-filter)# app ibm
ibm     ibm_as_central   ibm_as_dtaq  ibm_as_netprt ibm_as_srvmap ibm_iseries ibm_tsm
ibm_app ibm_as_database  ibm_as_file  ibm_as_rmtcmd ibm_db2       ibm_tealeaf
```

4. Filter applications by category using the **app-category** ***category name*** command. Currently, the applications contained in these categories are not displayed.

```
dmf-controller-1(config-managed-srv-app-id-filter)# app-category
<Category>          <String> :  <String>
aaa                 Category selection
adult_content       Category selection
advertising         Category selection
aetls               Category selection
analytics           Category selection
anonymizer          Category selection
audio_chat          Category selection
basic               Category selection
blog                Category selection
cdn                 Category selection
certif_auth         Category selection
chat                Category selection
classified_ads      Category selection
cloud_services      Category selection
crowdfunding        Category selection
cryptocurrency      Category selection
db                  Category selection
dea_mail            Category selection
ebook_reader        Category selection
education           Category selection
email               Category selection
enterprise          Category selection
file_mngt           Category selection
file_transfer       Category selection
forum               Category selection
gaming              Category selection
healthcare          Category selection
im_mc               Category selection
iot                 Category selection
map_service         Category selection
mm_streaming        Category selection
mobile              Category selection
networking          Category selection
news_portal         Category selection
p2p                 Category selection
payment_service     Category selection
remote_access       Category selection
scada               Category selection
social_network      Category selection
```

```
speedtest           Category selection
standardized        Category selection
transportation      Category selection
update              Category selection
video_chat          Category selection
voip                Category selection
vpn_tun             Category selection
web                 Category selection
web_ecom            Category selection
web_search          Category selection
web_sites           Category selection
webmail             Category selection
```

5. The **`filter-mode`** parameter supports two modes: forward and drop. Enter **`filter-mode forward`** to allow the packets to be forwarded based on the configured applications. Enter **`filter-mode drop`** to drop these packets.

   An example of an **`app-id-filter`** configuration that drops all Facebook and IBM Tealeaf packets:

```
managed-service MS
 service-interface switch CORE-SWITCH-1 ethernet2
 !
 1 app-id-filter
  app facebook
  app ibm_tealeaf
filter-mode drop
```



CAUTION: The app-id-filter configuration filters based on flows. For example, if a session is internally identified with the following tuple: ***ip***, ***tcp***, ***http***, ***google***, or ***google_maps***, adding any of these parameters to the filter list permits or drops all the packets matching after determining classification (e.g., adding ***tcp*** to the filter list permits or blocks packets from the aforementioned 5-tuple flow as well as all other tcp flows). Use caution when filtering using the lower-layer protocols and apps. Also, when forwarding an application, packets will be dropped at the beginning of the session until the application is identified. When dropping, packets at the beginning of the session will be passed until the application is identified.

5.19.4 Using the CLI to Configure app-id and app-id-filter Combined

Follow the configuration steps described in the services earlier to configure **app-id-filter** and **app-id** together. However, in this case, **app-id** should use a higher ***seq num*** than **app-id-filter**. Thus, the traffic is processed through the **app-id-filter** policy first, then through **app-id**.

This behavior can be helpful to monitor certain types of traffic. The following example illustrates a combined **app-id-filter** and **app-id** configuration.

```
! managed-service
managed-service MS1
service-interface switch CORE-SWITCH-1 ethernet2
!
!
1 app-id-filter
app facebook
filter-mode forward
!
2 app-id
collector 1.1.1.1
l3-delivery-interface L3-INTF-1
```



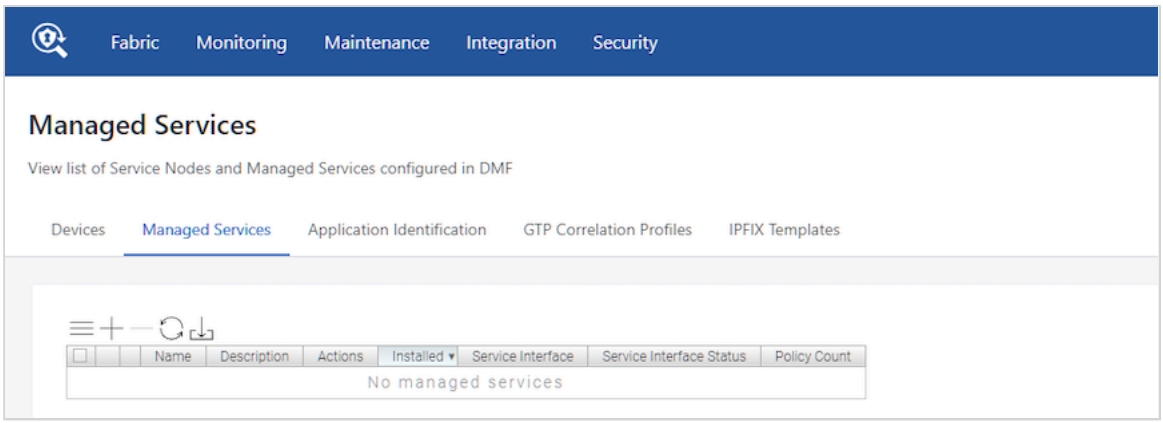
Note: The two drawbacks of this configuration are **app-id** dropping all traffic except *facebook*, and this type of service chaining can cause a performance hit and high memory utilization.

5.19.5 Using the GUI to Configure app-id and app-id-filter

App ID and **App ID Filter** are in the **Managed Service** workflow. Perform the following steps to complete the configuration.

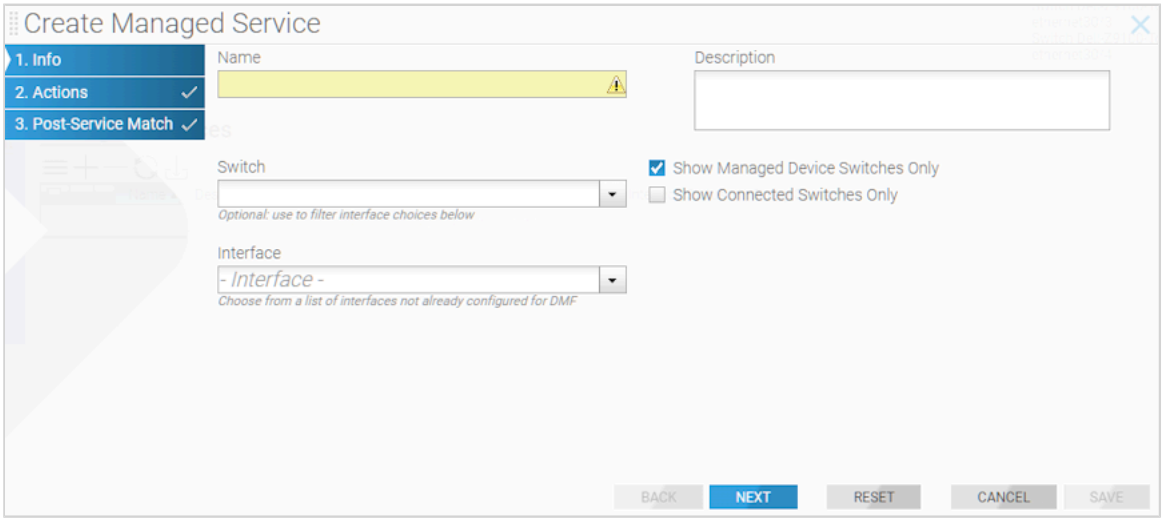
1. Navigate to the **Monitoring > Managed Services** page. Select the table action **+** icon to add a new managed service.

   Figure 5-56: DANZ Monitoring Fabric (DMF) Managed Services



2. Configure the **Name**, **Switch**, and **Interface** inputs in the **Info** step.

   Figure 5-57: Info Step



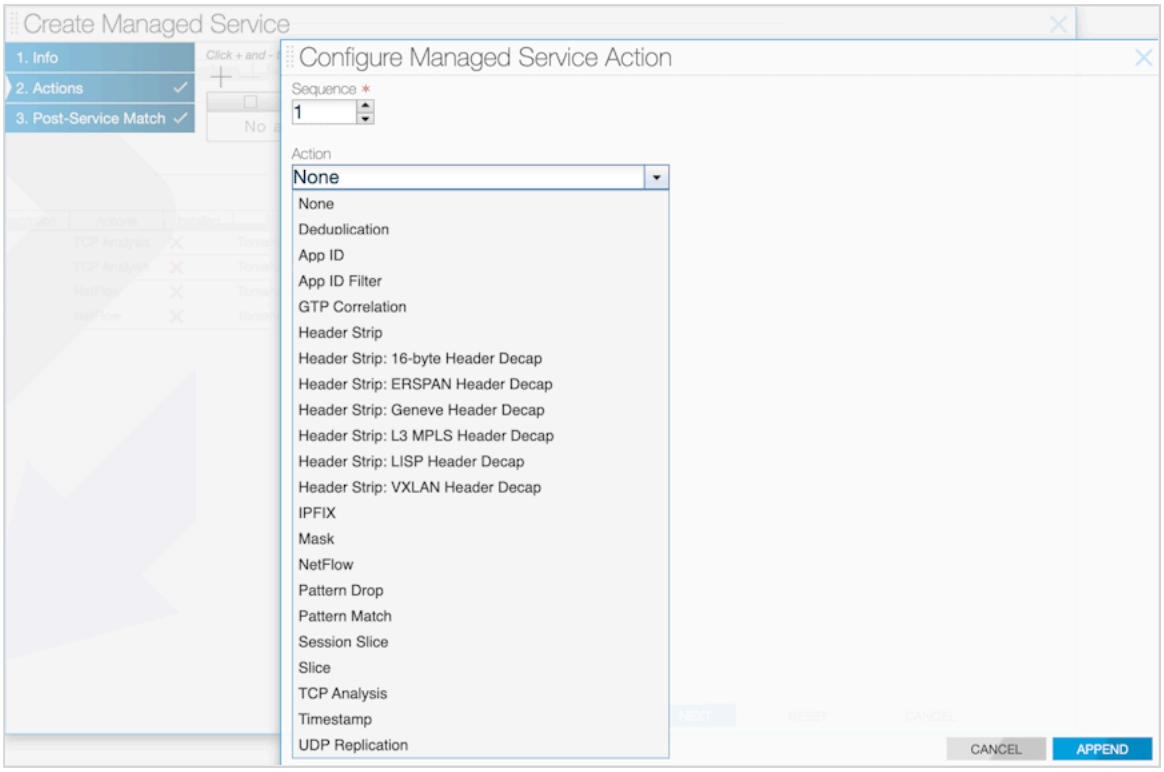
3. In the **Actions** step, select the **+** icon to add a new managed service action.

Figure 5-58: Add App ID Action



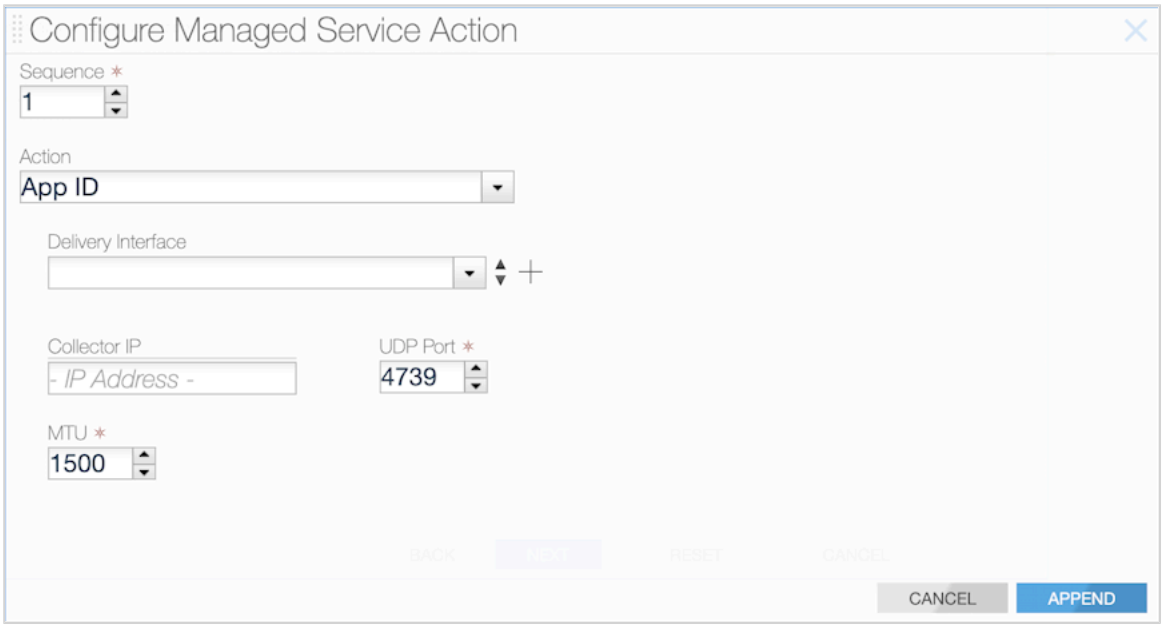
4. To Add the **App ID** Action, select **App ID** from the action selection input:

Figure 5-59: Select App ID



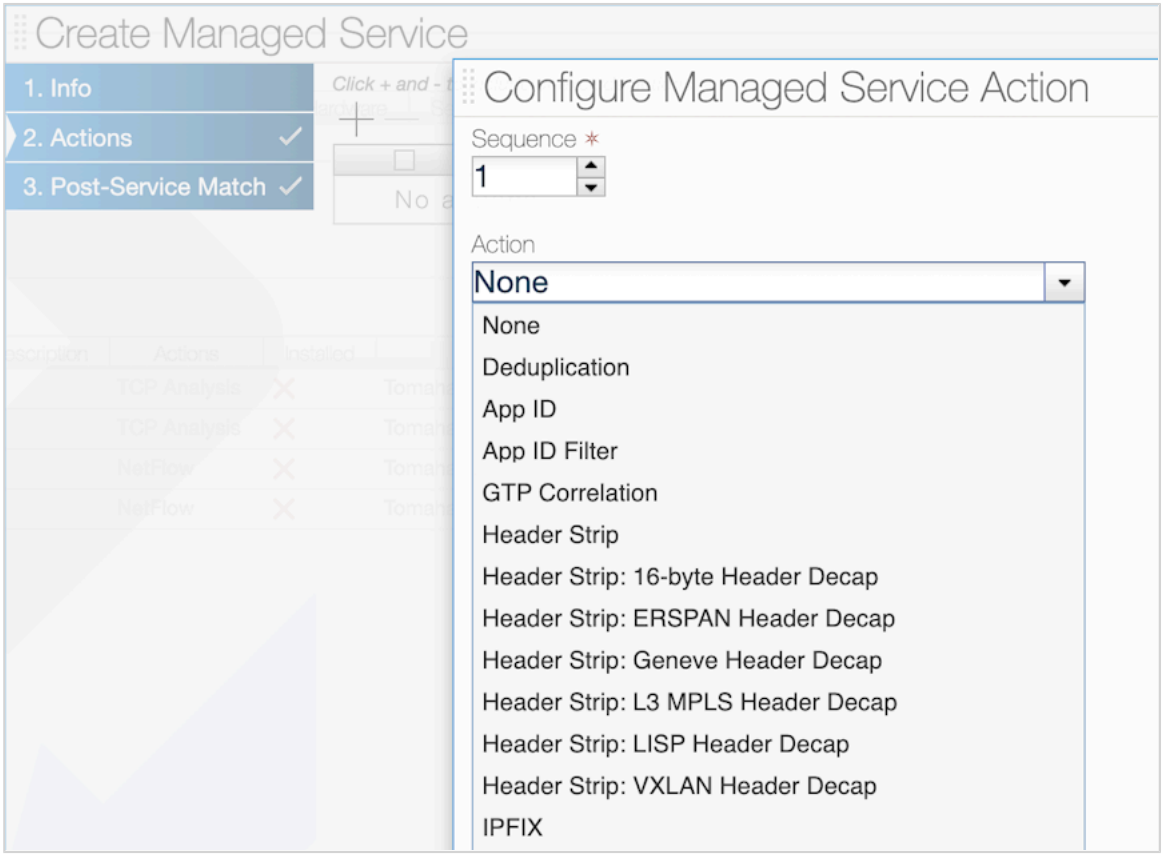
5. Fill in the **Delivery Interface**, **Collector IP**, **UDP Port**, and **MTU** inputs and select **Append** to include the action in the managed service:

Figure 5-60: Delivery Interface



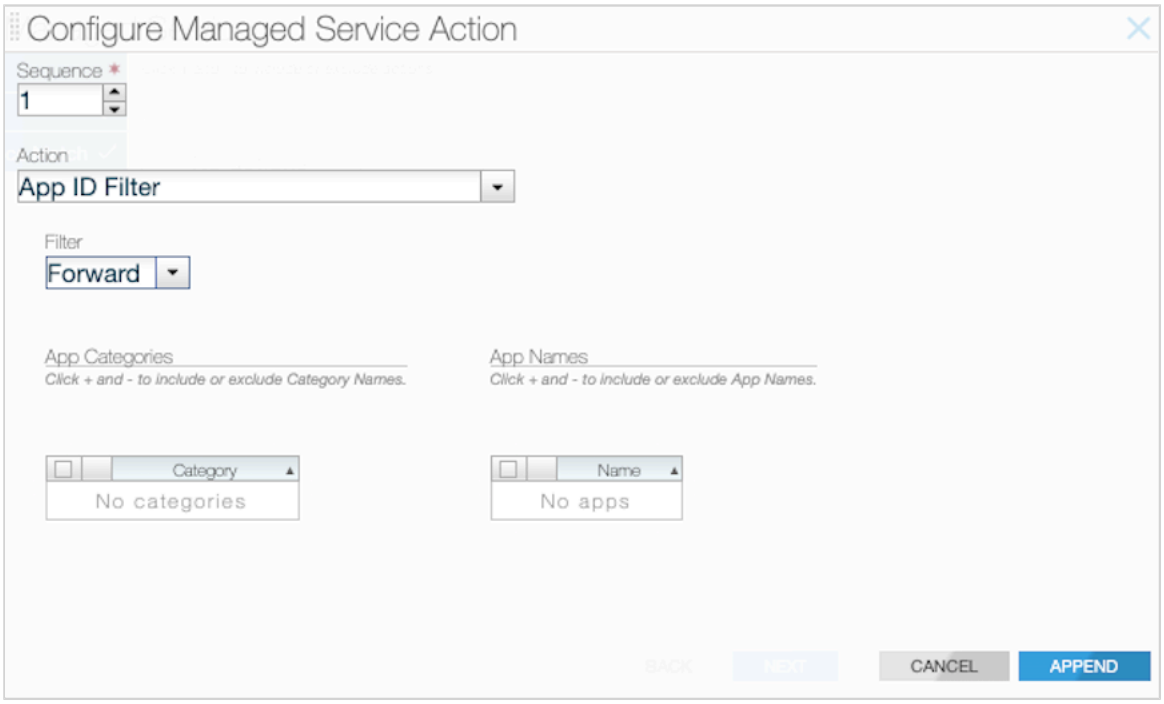
6. To Add the **App ID Filter** Action, select **App ID Filter** from the action selection input:

Figure 5-61: Select App ID Filter



7. Select the **Filter** input as **Forward** or **Drop** action:

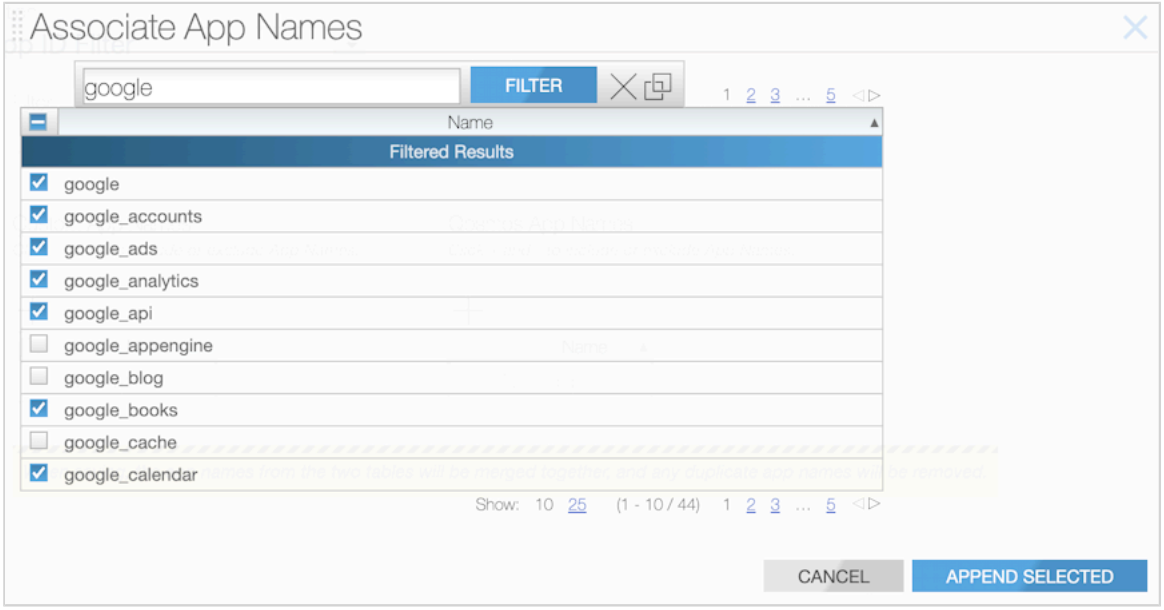
Figure 5-62: Select Filter Input



8. Use the **App Names** section to add app names.
   a. Select the **+** icon to open a modal pane to add an app name.
   b. The table lists all app names. Use the text search to filter out app names. Select the checkbox for app names to include and select **Append Selected**.

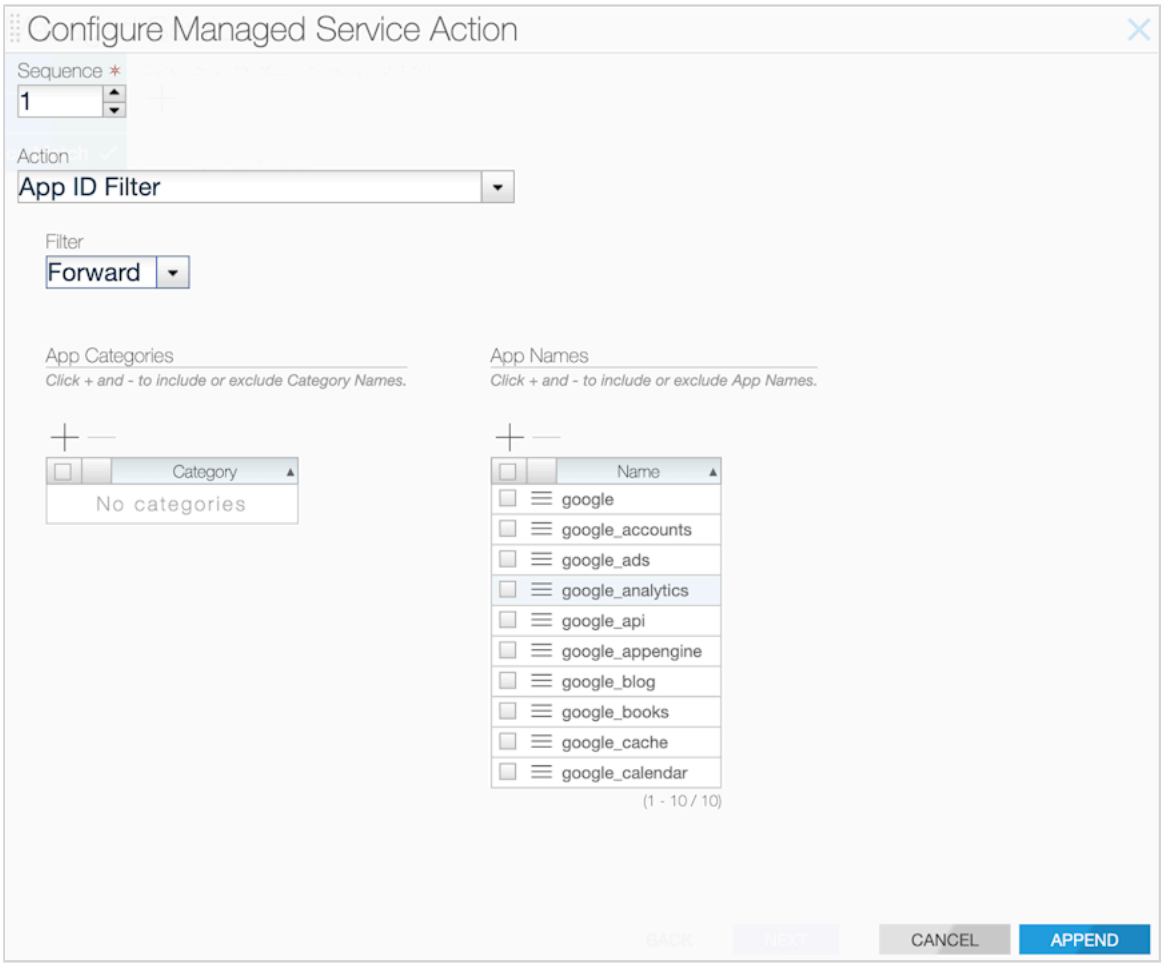
c. Repeat the above step to add more app names as necessary.

Figure 5-63: Associate App Names



9. The selected app names are now listed. Use the **-** icon to remove any app names, if necessary:

Figure 5-64: Application Names



10. Select **Append** to add the action to the managed service and **Save** to save the managed service.

For existing managed services, add **App ID** or **App ID Filter** using the **Edit** workflow of a managed service.

5.19.6 Dynamic Signature Updates (Beta Version)

This beta feature allows the app-id and app-id-filter services to classify newly supported applications at runtime rather than waiting for an update in the next DANZ Monitoring Fabric (DMF) release. Perform such runtime service updates during a maintenance cycle. There can be issues with backward compatibility if attempting to revert to an older bundle. Adopt only supported versions. In the Controller's CLI, perform the following recommended steps:

1. Remove all policies containing app-id or app-id-filter. Remove the app-id and app-id-filter managed services from the policies using the command: **`no use-managed-service`** in policy config.

   Arista Networks recommends this step to avoid errors and service node reboots during the update process. A warning message is printed right before confirming a push. Proceeding without this step may work but is not recommended as there is a risk of service node reboots.

   Note: Arista Networks provides the specific update file in the command example below.

2. To pull the signature file onto the Controller node, use the command:

```
dmf-controller-1(config)# app-id pull-signature-file user@host:path to file.tar.gz
Password:
file.tar.gz       5.47MB 1.63MBps 00:03
```

3. Fetch and validate the file using the command:

```
dmf-controller-1(config)# app-id fetch-signature-file file://file.tar.gz
Fetch successful.
Checksum   : abcdefgh12345
Fetch time : 2023-08-02 22:20:49.422000 UTC
Filename   : file.tar.gz
```

4. To view files currently saved on the Controller node after the fetch operation is successful, use the following command:

```
dmf-controller-1(config)# app-id list-signature-files
# Signature-file     Checksum     Fetch time
-|-----------------|-----------------|-----------------------------|
1 file.tar.gz   abcdefgh12345   2023-08-02 22:20:49.422000 UTC
```

   Note: Only the files listed by this command can be pushed to service nodes.

5. Push the file from the Controller to the service nodes using the following command:

```
dmf-controller-1(config)# app-id push-signature-file file.tar.gz
App ID update: WARNING: This push will affect all service nodes
App ID update: Remove policies configured with app-id or app-id-filter before continuing to
 avoid errors
App ID update: Signature file: file.tar.gz
App ID update: Push app ID signatures to all Service Nodes? Update ("y" or "yes" to continue):
 yes
Push successful.

Checksum     : abcdefgh12345
Fetch time   : 2023-08-02 22:20:49.422000 UTC
Filename     : file.tar.gz
Sn push time : 2023-08-02 22:21:49.422000 UTC
```

6. Add the **`app-id`** and **`app-id-filter`** managed services back to the policies.

   As a result of adding app-id, service nodes can now identify and report new applications to the analytics node.

After adding back app-id-filter, new application names should appear in the app-id-filter Controller app list. To test this, enter **`app-id-filter`** submode and press the **Tab** to see the full list of applications. New identified applications should appear in this list.

7. To delete a signature file from the Controller, use the command below.

> **Note:** DMF only allows deleting a signature file that is not actively in use by any service node, which needs to keep a working file in case of issues—attempting to delete an active file causes the command to fail.

```
dmf-controller-1(config)# app-id delete-signature-file file.tar.gz
Delete successful for file: file.tar.gz
```

Useful Information

The fetch and delete operations are synced with standby controllers as follows:

- **fetch**: after a successful fetch on the active Controller, it invokes the fetch RPC on the standby Controller by providing a signed HTTP URL as the source. This URL points to an internal REST API that provides the recently fetched signature file.
- **delete**: the active Controller invokes the delete RPC call on the standby controllers.

The Controller stores the signature files in this location: `/var/lib/capture/appidsignatureupdate`.

On a service node, files are overwritten and always contain the complete set of applications.

> **Note:** An analytics node cannot display these applications in the current version.

This step is only for informational purposes:

- Verify the bundle version on the service node by entering the **`show service-node app-id-bundle-version`** command in the service node CLI, as shown below.

Figure 5-65: Before Update

```
SD-appid84x-Hormigueros-SN1(config)# show service-node app-id-bundle-version
Name : bundle_version
Data : 1.640.2-24 (build date Mar  2 2023)
```

Figure 5-66: After Update

```
SD-appid84x-Hormigueros-SN1(config)# show service-node app-id-bundle-version
Name : bundle_version
Data : 1.650.2-22 (build date Apr 27 2023)
```

5.19.7 CLI Show Commands

Service Node

In the service node CLI, use the following **show** command:

```
show service-node app-id-bundle-version
```

This command shows the version of the bundle in use. An **app-id** or **app-id-filter** instance must be configured, or an error message is displayed.

```
dmf-servicenode-1# show app-id bundle-version
Name : bundle_version
Data : 1.680.0-22 (build date Sep 26 2023)
dmf-servicenode-1#
```

Controller

To obtain more information about the running version on a Service Node, or when the last push attempt was made and the outcome, use the following Controller CLI commands:

- **show app-id push-results *optional SN name***
- **show service-node *SN name* app-id**

```
dmf-controller-1# show app-id push-results
# Name              IP Address     Current Version Current Push Time              Previous Version
 Previous Push Time             Last Attempt Version Last Attempt Time              Last Attempt
 Result Last Attempt Failure Reason

-|-----------------|--------------|---------------|------------------------------|--------------
--|------------------------------|--------------------|------------------------------|-----
--------------|---------------------------|

1 dmf-servicenode-1 10.240.180.124 1.660.2-33      2023-12-06 11:13:36.662000 PST 1.680.0-22
 2023-09-29 16:21:11.034000 PDT 1.660.2-33           2023-12-06 11:13:34.085000 PST success
```

```
dmf-controller-1# show service-node dmf-servicenode-1 app-id
# Name              IP Address     Current Version Current Push Time              Previous Version
 Previous Push Time Last Attempt Version Last Attempt Time Last Attempt Result Last Attempt
 Failure Reason

-|-----------------|--------------|---------------|------------------------------|--------------
--|------------------|--------------------|-----------------|-------------------|----------
-----------------|

1 dmf-servicenode-1 10.240.180.124 1.680.0-22      2023-09-29 16:21:11.034000 PDT
```

The **show app-id signature-files** command displays the validated files that are available to push to Service Nodes.

```
dmf-controller-1# show app-id signature-files
# Signature-file    Checksum    Fetch time
-|-----------------|-----------------|------------------------------|
1 file1.tar.gz      abcdefgh12345     2023-08-02 22:20:49.422000 UTC
2 file2.tar.gz      ijklmnop67890     2023-08-03 07:10:22.123000 UTC
```

The **show analytics app-info *filter-interface-name*** command displays aggregated information over the last **5** minutes about the applications seen on a given filter interface, sorted by unique flow count. This command also has an optional **size** option to limit the number of results, default is all.



Note: This command only works in **push-per-filter** mode.

```
dmf-controller-1# show analytics app-info filter-interface f1 size 3
# App name Flow count
-|--------|----------|
1 app1     1000
2 app2     900
3 app3     800
```

5.19.8 Syslog Messages

Syslog messages for configuring the **app-id** and **app-id-filter** services appear in a service node's syslog through **journalctl**.

A Service Node syslog registers events for the **app-id** **add**, **modify**, and **delete** actions.

These events contain the keywords **dpi** and **dpi-filter**, which correspond to **app-id** and **app-id-filter**.

For example:

```
Adding dpi for port,
Modifying dpi for port,
Deleting dpi for port,
Adding dpi filter for port,
Modifying dpi filter for port,
Deleting dpi filter for port,
App appname does not exist - An invalid app name was entered.
```

The addition, modification, or deletion of app names in an **app-id-filter** managed-service in the Controller node's CLI influences the policy refresh activity, and these events register in **floodlight.log**.

5.19.9 Scale

- Max concurrent sessions are currently set to permit less than 200,000 active flows per core. Performance may drop the more concurrent flows there are. This value is a maximum value to prevent the service from overloading. Surpassing this threshold may cause some flows not to be processed, and the new flows will not be identified or filtered. Entries for inactive flows will time out after a few minutes for ongoing sessions and a few seconds after the session ends.
- If there are many inactive sessions, DMF holds the flow contexts, reducing the number of available flows used for DPI. The timeouts are approximately 7 minutes for TCP sessions and 1 minute for UDP.
- Heavy application traffic load degrades performance.

5.19.10 Troubleshooting

- If IPFIX reports do not appear on an Analytics Node (AN) or Collector, ensure the UDP port is configured correctly and verify the AN receives traffic.
- If the **app-id-filter** app list does not appear, ensure a Service Node (SN) is connected using the **show service-node** command on the Controller.
- For **app-id-filter**, enter at least one valid application from the list that appears using <Tab>. If not, the policy will fail to install with an error message app-id-filter `specified without at least one name TLV identifying application`.
- A flow may contain other IDs and protocols when using `app-id-filter`. For example, the specific application for a flow may be **google_maps**, but there may be protocols or broader applications under it, such as **ssh**, **http**, or **google**. Adding **google_maps** will filter this flow. However, adding **ssh** will also filter this flow. Therefore, adding any of these to the filter list will cause packets of this flow to be forwarded or dropped.
- An IPFIX element, BSN type 14, that existed in DMF version 8.4 was removed in 8.6.
- During a dynamic signature update, if a SN reboot occurs, it will likely boot up with the correct version. To avoid issues of traffic loss, perform the update during a maintenance window. Also, during an update, the SN will temporarily not send LLDP packets to the Controller and disconnect for a short while.
- After a dynamic signature update, do not change configurations or push another signature file for several minutes. The update will take some time to process. If there are any VFT changes, it may lead to warning messages in floodlight, such as:

```
Sync job 2853: still waiting after 50002 ms
Stuck switch update: R740-25G[00:00:e4:43:4b:bb:38:ca], duration=50002ms, stage=COMMIT
```

  These messages may also appear when configuring DPI on a large number of ports.

5.19.11 Limitations

- When using a drop filter, a few packets may slip through the filter before determining an application ID for a flow, and when using a forward filter, a few packets may not be forwarded. Such a small amount is estimated to be between 1 and 6 packets at the beginning of a flow.
- When using a drop filter, add the **unknown** app ID to the filter list to drop any unidentified traffic if these packets are unwanted.
- The Controller must be connected to a Service Node for the **app-id-filter** app list to appear. If the list does not appear and the application names are unknown, use the **app-id** to send reports to the analytics node. Use the application names seen there to configure an **app-id-filter**. The name must match exactly.
- Since **app-category** does not currently show the applications included in that category, do not use it when targeting specific apps. Categories like basic, which include all basic networking protocols like TCP and UDP, may affect all flows.
- For **app-id**, a report is only generated for a fully classified flow after that flow has been fully classified. Therefore, the number of reported applications may not match the total number of flows. These reports are sent after enough applications are identified on the Service Node. If many applications are identified, DMF sends the reports quickly. However, DMF sends these reports every 10 seconds when identifying only a few applications.
- DMF treats a bidirectional flow as part of the same n-tuple. As such, generated reports contain the client's source IP address and the server's destination IP address.
- While configuring many ports with the **app-id**, there may occasionally be a few Rx drops on the 16 port machines at a high traffic rate in the first couple of seconds.
- The feature uses a cache that maps dest ip and port to the application. Caching may vary the performance depending on the traffic profile.
- The **app-id** and **app-id-filter** services are more resource-intensive than other services. Combining them in a service chain or configuring many instances of them may lead to degradation in performance.
- At scale, such as configuring 16 ports on the R740 DCA-DM-SEL, **app-id** may take a few minutes to set up on all these ports, and this is also true when doing a dynamic signature update.
- The **show analytics app-info** command only works in **push-per-filter** VLAN mode.
- Mapping an interface-name with identified application IDs from source traffic received via the filter interface at Analytics Dashboard is allowed in Push-Per-Filter Mode only.

5.20 Redundancy of Managed Services Using Two DMF Policies

In this method, users can employ a second policy with a second managed service to provide redundancy. The idea here is to duplicate the policies but assign a lower policy priority to the second DANZ Monitoring Fabric (DMF) policy. In this case, the backup policy (and, by extension, the backup service) will always be active but only receive relevant traffic once the primary policy goes down. This method provides true redundancy at the policy, service-node, and core switch levels but uses additional network and node resources.

Example

```
! managed-service
managed-service MS-SLICE-1
1 slice l3-header-start 20
service-interface switch CORE-SWITCH-1 lag1
!
managed-service MS-SLICE-2
1 slice l3-header-start 20
service-interface switch CORE-SWITCH-1 lag2
! policy
policy ACTIVE-POLICY
priority 101
action forward
delivery-interface TOOL-PORT-1
filter-interface TAP-PORT-1
use-managed-service MS-SLICE-1 sequence 1
1 match ip
!
policy BACKUP-POLICY
priority 100
action forward
delivery-interface TOOL-PORT-1
filter-interface TAP-PORT-1
use-managed-service MS-SLICE-2 sequence 1
1 match ip
```

5.21 Sharing Managed Services Across Policies

A DANZ Monitoring Fabric (DMF) Controller allows sharing of managed services utilizing L3 delivery interfaces (e.g., NetFlow, IPFIX, app ID, etc.) across multiple policies. Prior to the DMF 8.7.0 release, DMF did not support managed service sharing because the L3 delivery interface was an optional setting in a policy configuration. However, sharing is now supported because the managed service configuration must now specify the L3 delivery interface.

When multiple policies (overlapping or non-overlapping) share the same Managed Service, the system will ensure that post-service traffic from all these policies is forwarded to appropriate delivery interfaces using a dynamically created post-service policy.

This change applies to the following managed service actions using L3 deliveries on all platforms that support managed services:

- Netflow
- IPFIX
- UDP Replicate
- TCP Analysis
- App ID
- Flow Diff

5.21.1 Configuration using the CLI

There are no new configurations in the Sharing Managed Services Across Policies feature, but several configuration validations surrounding the managed service configuration within a policy have changed:

- Removed the validation that prevented adding an L3 header modifying managed service to multiple policies.
- A validation was added to enforce the order of shared managed services.

When sharing multiple managed services with at least one action that requires an L3 delivery interface, the only valid case is where **non-UDP replicate** L3 services are followed by **UDP replicate**:

- Invalid: any **non UDP replicate** → any **non UDP replicate**.
- Invalid: **UDP replicate** → any **non UDP replicate**.
- Invalid: **UDP replicate** → **UDP replicate**.
- Valid: any **non UDP replicate** → **UDP replicate**.

There are two fundamental exceptions to the earlier examples:

- The **push-per-filter** mode doesn't allow multiple header modifying services in a policy.
- **Flow diff** is only supported in **push-per-filter** mode and cannot be chained with **UDP replicate**.

The following is a sample configuration sharing a Netflow managed service across two policies.

Configure an L3 delivery interface:

```
dmf-controller> enable
dmf-controller# configure
dmf-controller(config)# switch sw1
dmf-controller(config-switch)# interface ethernet31
dmf-controller(config-switch-if)# role delivery interface-name l3-d1 ip-address 0.0.0.1 nexthop-ip
 0.0.0.1 255.255.255.0 nexthop-arp-interval 5
dmf-controller(config-switch-if)# rate-limit 1000
```

Create two managed services, one that is non-shareable and one with **netflow**, which can be shared:

Non-shared Managed Service

```
dmf-controller> enable
dmf-controller# configure
dmf-controller(config)# managed-service ms-non-shareable
dmf-controller(config-managed-srv)# service-interface switch sw1 ethernet2
dmf-controller(config-managed-srv)# 1 sample
dmf-controller(config-managed-srv-sample)# max-tokens 100
dmf-controller(config-managed-srv-sample)# tokens-per-refresh 10
```

Shared Managed Service

```
dmf-controller> enable
dmf-controller# configure
dmf-controller(config)# managed-service ms-netflow
dmf-controller(config-managed-srv)# service-interface switch sw1 ethernet1
dmf-controller(config-managed-srv)# 1 netflow
dmf-controller(config-managed-srv-netflow)# collector 0.0.0.2 udp-port 2055 mtu 1500
dmf-controller(config-managed-srv-netflow)# l3-delivery-interface l3-d1
```

Configure two policies, where the first policy, **p1**, has both **ms-non-shareable** and **ms-netflow** in order, and the second policy, **p2**, only has **ms-netflow**:

Policy 1

```
dmf-controller> enable
dmf-controller# configure
dmf-controller(config)# managed-service policy p1
dmf-controller(config-policy)# action forward
dmf-controller(config-policy)# 1 match any
dmf-controller(config-policy)# filter-interface f1
dmf-controller(config-policy)# use-managed-service ms-non-shareable sequence 1
dmf-controller(config-policy)# use-managed-service ms-netflow sequence 2
```

Policy 2

```
dmf-controller> enable
dmf-controller# configure
dmf-controller(config)# managed-service policy p2
dmf-controller(config-policy)# action forward
dmf-controller(config-policy)# 1 match any
dmf-controller(config-policy)# filter-interface f2
dmf-controller(config-policy)# use-managed-service ms-netflow sequence 1
```

In **push-per-policy mode**, both **p1** and **p2** are expected to be installed to send traffic to the **ms-netflow** pre-service interface, and a dynamic post-service policy **p1_p2__post_to_delivery__** is created to carry the traffic from the post-service interface of **ms-netflow** to **l3-d1**.

In **push-per-filter** mode, a post service policy is created regardless of whether a managed service is shared or not. The result is two post service policies, **p1__post_to_delivery__** and **p2__post_to_delivery__**, which will carry traffic from their respective configured policies to the delivery interface via the configured services.

The following section shows the runtime state of these policies.

5.21.2 Show Commands

While no new **show** commands accompany the Sharing Managed Services Across Policies feature, use the following existing commands to verify the configuration.

Use the **show running-config switch sw1 interface *l3 interface*** command to view the configured L3 interface:

```
dmf-controller> show running-config switch sw1 interface ethernet31

! switch
switch sw1
  !
  interface ethernet31
    force-link-up
    rate-limit 1000
    role delivery interface-name l3-d1 ip-address 0.0.0.1 nexthop-ip 0.0.0.2 255.255.255.0
 nexthop-arp-interval 5
```

Use the **show running-config managed-service** command to view the managed service running config:

```
dmf-controller> show running-config managed-service

! managed-service
managed-service ms-netflow
  service-interface switch sw1 ethernet1
  !
  1 netflow
    collector 10.0.0.2 udp-port 2055 mtu 1500
    l3-delivery-interface l3-d1

managed-service ms-non-shareable
  service-interface switch sw1 ethernet2
  !
  1 sample
    max-tokens 100
    tokens-per-refresh 10
```

Use the **show running-config policy** to view the policy running config:

```
dmf-controller> show running-config policy

! policy
policy p1
  action forward
  filter-interface f1
  use-managed-service ms-netflow sequence 2
  use-managed-service ms-non-shareable sequence 1
  1 match any

policy p2
  action forward
  filter-interface f2
  use-managed-service ms-netflow sequence 1
  1 match any
```

Use the **show switch sw1 interface *l3 interface*** command to view the runtime status of the interface:

```
dmf-controller> show switch sw1 interface ethernet31
# IF Name    MAC Address               Config State Adv. Features Curr Features Supported
 Features
-|----------|-------------------------|------|-----|-------------|-------------|--------------
----|
1 ethernet31 5c:16:c7:14:46:bb (Arista) up     up    10g           10g           10g
```

Use the **show managed-service** command to view the runtime status of the managed services:

```
dmf-controller># show managed-service
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Managed-service ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~
# Service Name     Switch Switch Interface Installed Max Post-Service BW Max Pre-Service BW Total
 Post-Service BW Total Pre-Service BW
-|----------------|------|----------------|---------|-------------------|---------------
---|--------------------|--------------------|
1 ms-netflow       sw1    ethernet1        True      10Gbps              10Gbps             974bps
                66bps
2 ms-non-shareable sw1    ethernet2        True      10Gbps              10Gbps             960bps
                66bps
```

Push-per-policy Mode

The **show policy** command displays a brief version of runtime state of all policies, including those created dynamically:

```
dmf-controller> show policy
# Policy Name               Action  Runtime Status Type       Priority Overlap Priority Push VLAN
 Filter BW Delivery BW Post Match Filter Traffic Delivery Traffic Services         Installed Time
        Installed Duration  Ptp Timestamping
-|-------------------------|-------|--------------|----------|--------|----------------|---
------|---------|-----------|-------------------------|----------------|----------------|--
---------------------|--------------------|----------------|
1 p1                        forward installed      Configured 100      0                1
   10Gbps    10Gbps      -                         -                ms-non-shareable 2025-02-20
 18:18:33 UTC 20 minutes, 46 secs False
2 p2                        forward installed      Configured 100      0                2
   10Gbps    10Gbps      -                         -                                 2025-02-20
 18:18:33 UTC 20 minutes, 46 secs False
3 p1_p2__post_to_delivery__ forward installed      Dynamic    100      0                3
   10Gbps    10Gbps      -                         -                                 2025-02-20
 18:18:33 UTC 20 minutes, 46 secs False
```

The **show policy *policy name*** command provides a more detailed view of a policy at runtime.

Since a new post-service policy is required for shared services in **push-per-policy** mode (in the **push-per-filter** mode, a post-service policy, shared or not, is always created), the configured policies are adjusted accordingly. So, **p1** and **p2** will neither have a **ms-netflow** service nor will **l3-d1** be the delivery interface.

p1 carries traffic from **f1** to **ms-non-shareable** and then delivers it to **ms-netflow**.

```
dmf-controller> show policy p1
Policy Name                             : p1
Config Status                           : active - forward
Runtime Status                          : installed
Detailed Status                         : installed - installed to forward
Priority                                : 100
Overlap Priority                        : 0
# of switches with filter interfaces    : 1
# of switches with delivery interfaces  : 1
# of switches with service interfaces   : 1
# of filter interfaces                  : 1
# of delivery interfaces                : 1
# of core interfaces                    : 0
# of services                           : 1
# of pre service interfaces             : 1
# of post service interfaces            : 1
Push VLAN                               : 1
Post Match Filter Traffic               : -
Total Delivery Rate                     : -
Total Pre Service Rate                  : -
Total Post Service Rate                 : -
Overlapping Policies                    : p1_p2__post_to_delivery__,
Component Policies                      : none
Runtime Service Names                   : ms-non-shareable
Installed Time                          : 2025-02-20 18:18:33 UTC
Installed Duration                      : 23 minutes, 18 secs
Timestamping enabled                    : False
~ Match Rules ~
# Rule
-|-----------|
1 1 match any

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF Switch IF Name    State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time
-|------|------|----------|-----|---|-------|-----|--------|--------|------------------|
1 f1     sw1    ethernet11 up    rx  0       0     0        -

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~
# DMF IF                            Switch IF Name   State Dir Packets Bytes Pkt Rate Bit Rate
 Counter Reset Time
-|---------------------------------|------|---------|-----|---|-------|-----|--------
|--------|------------------|
1 sw1  -ethernet1-to-managed-service sw1    ethernet1 up    tx  2       140   0        -

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Service Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~
# Service name     Role         Switch IF Name   State Dir Packets Bytes Pkt Rate Bit Rate Counter
 Reset Time
-|----------------|------------|------|---------|-----|---|-------|-----|--------|----
----|------------------|
1 ms-non-shareable pre-service  sw1    ethernet2 up    tx  0       0     0        -
2 ms-non-shareable post-service sw1    ethernet2 up    rx  2       140   0        -

~ Core Interface(s) ~
None.

~ Failed Path(s) ~
None.
```

p2 carries traffic from **f2** to **ms-netflow**.

```
dmf-controller> show policy p2
Policy Name                             : p2
Config Status                           : active - forward
Runtime Status                          : installed
Detailed Status                         : installed - installed to forward
Priority                                : 100
Overlap Priority                        : 0
# of switches with filter interfaces    : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces  : 0
# of filter interfaces                  : 1
# of delivery interfaces                : 1
# of core interfaces                    : 0
# of services                           : 0
# of pre service interfaces             : 0
# of post service interfaces            : 0
Push VLAN                               : 2
Post Match Filter Traffic               : -
Total Delivery Rate                     : -
Total Pre Service Rate                  : -
Total Post Service Rate                 : -
Overlapping Policies                    : p1_p2__post_to_delivery__,
Component Policies                      : none
Installed Time                          : 2025-02-20 18:18:33 UTC
Installed Duration                      : 23 minutes, 21 secs
Timestamping enabled                    : False
~ Match Rules ~
# Rule
-|-----------|
1 1 match any

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF Switch IF Name    State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time
-|------|------|----------|-----|---|-------|-----|--------|--------|------------------|
1 f2     sw1    ethernet12 up    rx  0       0     0        -

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~
# DMF IF                              Switch IF Name   State Dir Packets Bytes Pkt Rate Bit Rate
 Counter Reset Time
-|-----------------------------------|------|---------|-----|---|-------|-----|--------
|--------|------------------|
1 sw1-ethernet1-to-managed-service   sw1    ethernet1 up    tx  0       0     0        -

~ Service Interface(s) ~
None.

~ Core Interface(s) ~
None.

~ Failed Path(s) ~
None.
```

p1_p2__post_to_delivery__ receives traffic from **ms-netflow**, and then delivers it to **l3-d1**.

```
dmf-controller> show policy p1_p2__post_to_delivery__
Policy Name                             : p1_p2__post_to_delivery__
Config Status                           : active - forward
Runtime Status                          : installed
Detailed Status                         : installed - installed to forward
Priority                                : 100
Overlap Priority                        : 0
# of switches with filter interfaces    : 1
# of switches with delivery interfaces  : 1
# of switches with service interfaces   : 0
# of filter interfaces                  : 1
# of delivery interfaces                : 1
# of core interfaces                    : 0
# of services                           : 0
# of pre service interfaces             : 0
# of post service interfaces            : 0
Push VLAN                               : 3
Post Match Filter Traffic               : -
Total Delivery Rate                     : -
Total Pre Service Rate                  : -
Total Post Service Rate                 : -
Overlapping Policies                    : none
Component Policies                      : p1, p2,
Installed Time                          : 2025-02-20 18:18:33 UTC
Installed Duration                      : 23 minutes, 28 secs
Timestamping enabled                    : False
~ Match Rules ~
None.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~
# DMF IF                              Switch IF Name   State Dir Packets Bytes Pkt Rate Bit Rate
 Counter Reset Time
-|-----------------------------------|------|---------|-----|---|-------|-----|--------
|--------|------------------|
1 sw1-ethernet1-to-managed-service   sw1    ethernet1 up    rx  1       70    0        -

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF Switch IF Name    State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time
-|------|------|----------|-----|---|-------|-----|--------|--------|------------------|
1 l3-d1  sw1    ethernet31 up    tx  1       70    0        -

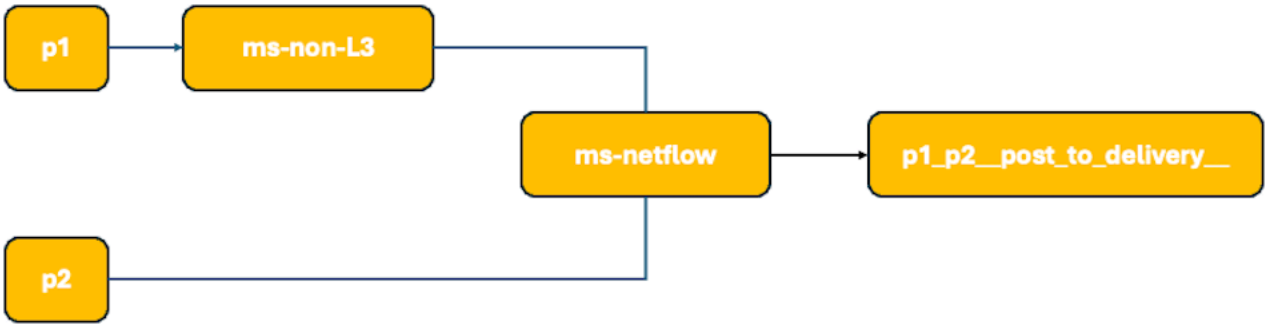
~ Service Interface(s) ~
None.

~ Core Interface(s) ~
None.

~ Failed Path(s) ~
None.
```

Another way of visualizing this runtime state is illustrated in the following diagram:

Figure 5-67: Example - Runtime State



Push-per-filter Mode

The **`show policy`** command displays a brief version of the runtime state of all policies, including any created dynamically:

```
dmf-controller> show policy
# Policy Name            Action  Runtime Status                    Type       Priority Overlap
 Priority Push VLAN Filter BW Delivery BW Post Match Filter Traffic Delivery Traffic Services
 Installed Time          Installed Duration  Ptp Timestamping
-|----------------------|-------|---------------------------------|----------|--------|----
------------|---------|---------|-----------|-------------------------|----------------|---
-------|----------------------|------------------|----------------|
1 p1                     forward installed                         Configured 100      0
          0         10Gbps    10Gbps      -                         -
 2025-02-20 19:13:39 UTC 15 minutes, 29 secs False
2 p2                     forward installed                         Configured 100      0
          0         10Gbps    10Gbps      -                         -
 2025-02-20 19:13:39 UTC 15 minutes, 29 secs False
3 p1__post_to_delivery__ forward installed                         Dynamic    100      0
          0         10Gbps    10Gbps      -                         -                ms-netflow
 2025-02-20 19:13:39 UTC 15 minutes, 29 secs                                            False
4 p2__post_to_delivery__ forward installed                         Dynamic    100      0
          0         10Gbps    10Gbps      -                         -
 2025-02-20 19:13:39 UTC 15 minutes, 29 secs False
```

The **`show policy`** ***`policy name`*** command provides a more detailed view of a policy at runtime:

p1 carries traffic from **f1** to **ms-non-shareable** service.

```
dmf-controller> show policy p1
Policy Name                           : p1
Config Status                         : active - forward
Runtime Status                        : installed
Detailed Status                       : installed - installed to forward
Priority                              : 100
Overlap Priority                      : 0
# of switches with filter interfaces   : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces  : 0
# of filter interfaces                 : 1
# of delivery interfaces               : 1
# of core interfaces                   : 0
# of services                          : 0
# of pre service interfaces            : 0
# of post service interfaces           : 0
Push VLAN                             : 0
Post Match Filter Traffic             : -
Total Delivery Rate                   : -
Total Pre Service Rate                : -
Total Post Service Rate               : -
Overlapping Policies                  : p1__post_to_delivery__,
Component Policies                    : none
Installed Time                        : 2025-02-20 19:13:39 UTC
Installed Duration                    : 17 minutes, 55 secs
Timestamping enabled                  : False
~ Match Rules ~
# Rule
-|-----------|
1 1 match any

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF Switch IF Name    State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time
-|------|------|----------|-----|---|-------|-----|--------|--------|------------------|
1 f1     sw1    ethernet11 up    rx  0       0     0        -

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~
# DMF IF                             Switch IF Name   State Dir Packets Bytes Pkt Rate Bit Rate
 Counter Reset Time
-|----------------------------------|------|---------|-----|---|-------|-----|--------
|--------|------------------|
1 sw1-ethernet2-to-managed-service   sw1    ethernet2 up    tx  0       0     0        -

~ Service Interface(s) ~
None.

~ Core Interface(s) ~
None.

~ Failed Path(s) ~
None.
```

p2 carries traffic from **f1** to **ms-netflow** service.

```
dmf-controller> show policy p2
Policy Name                             : p2
Config Status                           : active - forward
Runtime Status                          : installed
Detailed Status                         : installed - installed to forward
Priority                                : 100
Overlap Priority                        : 0
# of switches with filter interfaces    : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces  : 0
# of filter interfaces                  : 1
# of delivery interfaces                : 1
# of core interfaces                    : 0
# of services                           : 0
# of pre service interfaces             : 0
# of post service interfaces            : 0
Push VLAN                               : 0
Post Match Filter Traffic               : -
Total Delivery Rate                     : -
Total Pre Service Rate                  : -
Total Post Service Rate                 : -
Overlapping Policies                    : p2__post_to_delivery__,
Component Policies                      : none
Installed Time                          : 2025-02-20 19:13:39 UTC
Installed Duration                      : 18 minutes, 44 secs
Timestamping enabled                    : False
~ Match Rules ~
# Rule
-|-----------|
1 1 match any

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF Switch IF Name    State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time
-|------|------|----------|-----|---|-------|-----|--------|--------|------------------|
1 f2     sw1    ethernet12 up    rx  0       0     0        -

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~
# DMF IF                              Switch IF Name   State Dir Packets Bytes Pkt Rate Bit Rate
 Counter Reset Time
-|-----------------------------------|------|---------|-----|---|-------|-----|--------
|--------|------------------|
1 sw1-ethernet1-to-managed-service   sw1    ethernet1 up    tx  0       0     0        -

~ Service Interface(s) ~
None.

~ Core Interface(s) ~
None.

~ Failed Path(s) ~
None.
```

p1__post_to_delivery__ carries traffic from **ms-non-shareable** to **ms-netflow** and then to **l3-d1**.

```
dmf-controller> show policy p1__post_to_delivery__
Policy Name                           : p1__post_to_delivery__
Config Status                         : active - forward
Runtime Status                        : installed
Detailed Status                       : installed - installed to forward
Priority                              : 100
Overlap Priority                      : 0
# of switches with filter interfaces   : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces  : 1
# of filter interfaces                 : 1
# of delivery interfaces               : 1
# of core interfaces                   : 0
# of services                          : 1
# of pre service interfaces            : 1
# of post service interfaces           : 1
Push VLAN                             : 0
Post Match Filter Traffic             : -
Total Delivery Rate                   : -
Total Pre Service Rate                : -
Total Post Service Rate               : -
Overlapping Policies                  : none
Component Policies                    : p1,
Runtime Service Names                 : none
Timestamping enabled                  : False
~ Match Rules ~
None.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~
# DMF IF                              Switch IF Name   State Dir Packets Bytes Pkt Rate Bit Rate
 Counter Reset Time
-|----------------------------------|------|---------|-----|---|-------|-----|--------
|--------|------------------|
1 sw1-ethernet2-to-managed-service   sw1    ethernet2 up    rx  0       0     0        -

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF Switch IF Name    State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time
-|------|------|----------|-----|---|-------|-----|--------|--------|------------------|
1 l3-d1  sw1    ethernet31 up    tx  0       0     0        -

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Service Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~
# Service name Role         Switch IF Name   State Dir Packets Bytes Pkt Rate Bit Rate Counter
 Reset Time
-|------------|------------|------|---------|-----|---|-------|-----|--------|--------
|------------------|
1 ms-netflow   pre-service  sw1    ethernet1 up    tx  0       0     0        -
2 ms-netflow   post-service sw1    ethernet1 up    rx  0       0     0        -

~ Core Interface(s) ~
None.

~ Failed Path(s) ~
None.
```

p2__post_to_delivery__ carries traffic from **ms-netflow** to **l3-d1**.

```
dmf-controller> show policy p2__post_to_delivery__
Policy Name                             : p2__post_to_delivery__
Config Status                           : active - forward
Runtime Status                          : installed
Detailed Status                         : installed - installed to forward
Priority                                : 100
Overlap Priority                        : 0
# of switches with filter interfaces    : 1
# of switches with delivery interfaces  : 1
# of switches with service interfaces   : 0
# of filter interfaces                  : 1
# of delivery interfaces                : 1
# of core interfaces                    : 0
# of services                           : 0
# of pre service interfaces             : 0
# of post service interfaces            : 0
Push VLAN                               : 0
Post Match Filter Traffic               : -
Total Delivery Rate                     : -
Total Pre Service Rate                  : -
Total Post Service Rate                 : -
Overlapping Policies                    : none
Component Policies                      : p2,
Installed Time                          : 2025-02-20 19:13:39 UTC
Installed Duration                      : 22 minutes, 19 secs
Timestamping enabled                    : False
~ Match Rules ~
None.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~
# DMF IF                                Switch IF Name   State Dir Packets Bytes Pkt Rate Bit Rate
 Counter Reset Time
-|-----------------------------------|------|---------|-----|---|-------|-----|--------
|--------|------------------|
1 sw1-ethernet1-to-managed-service   sw1    ethernet1 up    rx  0       0     0        -

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF Switch IF Name    State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time
-|------|------|----------|-----|---|-------|-----|--------|--------|------------------|
1 l3-d1  sw1    ethernet31 up    tx  0       0     0        -

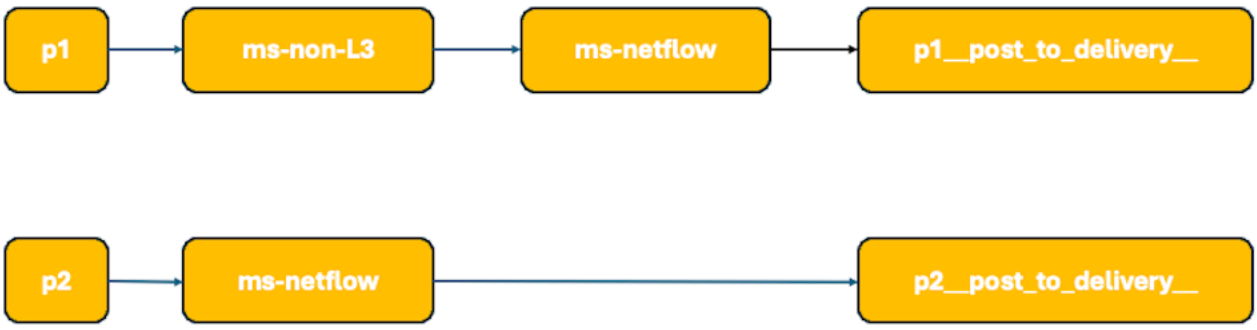
~ Service Interface(s) ~
None.

~ Core Interface(s) ~
None.

~ Failed Path(s) ~
None.
```

Another way of visualizing this runtime state is illustrated in the following diagram:

Figure 5-68: Example - Runtime State



5.21.3 Troubleshooting

If any failures occur by the sharing of L3 managed services between policies, ensure the order of services shared by all policies is the same, i.e., the order of services matters and partially sharing the sequence is not allowed:

- Policy **p1** cannot have **ms1** → **ms2**, while policy **p2** already has **ms2** → **ms1**.
- Policy **p1** cannot have **ms1** → **ms2**, while policy **p2** only has **ms1**.

5.21.4 Limitations

- When creating a post-service policy, configured policies are modified at runtime. So, the runtime state of a configured policy will show the modified delivery interfaces and services. You must use configured and dynamic policies to visualize the complete path.
- Not all managed service actions can be shared.
- Shared managed services must be the last in the sequence, and in the same order for all policies using it. Partial sharing of a sub-sequence of services is not allowed. Policy **p1** cannot have **ms1** → **ms2**, while policy **p2** only has **ms1**.

5.22 Cloud Services Filtering

The DANZ Monitoring Fabric (DMF) supports traffic filtering to specific services hosted in the public cloud and redirecting filtered traffic to customer tools. DMF achieves this functionality by reading the source and destination IP addresses of specific flows, identifying the Autonomous System number they belong to, tagging the flows with their respective AS numbers, and redirecting them to customer tools for consumption.

The following is the list of services supported:

- **amazon**: traffic with src/dst IP belonging to Amazon
- **ebay**: traffic with src/dst IP belonging to eBay
- **facebook**: traffic with src/dst IP belonging to FaceBook
- **google**: traffic with src/dst IP belonging to Google
- **microsoft**: traffic with src/dst IP belonging to Microsoft
- **netflix**: traffic with src/dst IP belonging to Netflix
- **office365**: traffic for Microsoft Office365
- **sharepoint**: traffic for Microsoft Sharepoint
- **skype**: traffic for Microsoft Skype
- **twitter**: traffic with src/dst IP belonging to Twitter
- **default**: traffic not matching other rules in this service. Supported types are **match** or **drop**.

The option **drop** instructs the DMF Service Node to drop packets matching the configured application.

The option **match** instructs the DMF Service Node to deliver packets to the delivery interfaces connected to the customer tool.

A default drop action is auto-applied as the last rule, except when configuring the last rule as **`match default`**. It instructs the DMF Service Node to drop packets when either of the following conditions occurs:

- The stream's source IP address or destination IP address doesn't belong to any AS number.
- The stream's source IP address or destination IP address is affiliated with an AS number but has no specific action set.

5.22.1 Cloud Services Filtering Configuration

Managed Service Configuration

```
Controller(config)# managed-service name
Controller(config-managed-srv)#
```

Service Action Configuration

```
Controller(config-managed-srv)# 1 app-filter
Controller(config-managed-srv-appfilter)#
```

Filter Rules Configuration

```
Controller(config-managed-srv-appfilter)# 1 drop sharepoint
Controller(config-managed-srv-appfilter)# 2 match google
Controller(config-managed-srv-appfilter)# show this
! managed-service
managed-service sf3
service-interface switch CORE-SWITCH-1 ethernet13/1
!
1 service- app-filter
1 drop sharepoint
2 match google
```

A policy having a managed service with app-filter as the managed service, but with no matches specified will fail to install. The example below shows a **`policy incomplete-policy`** having failed due to the absence of a Match/Drop rule in the managed service incomplete-managed-service.

```
Controller(config)# show running-config managed-service incomplete-managed-service
! managed-service
managed-service incomplete-managed-service
1 app-filter
Controller(config)# show running-config policy R730-sf3
! policy
policy incomplete-policy
action forward
delivery-interface TOOL-PORT-1
filter-interface TAP-PORT-1
use-managed-service incomplete-managed-service sequence 1
1 match any
```

```
Controller(config-managed-srv-appfilter)# show policy incomplete-policy
Policy Name : incomplete-policy
Config Status : active - forward
Runtime Status : one or more required service down
Detailed Status : one or more required service down - installed to
forward
Priority : 100
Overlap Priority : 0
```

5.23 Multiple Services Per Service Node Interface

The service-node capability is augmented to support more than one service action per service-node interface. Though this feature is economical regarding per-interface cost, it could cause packet drops in high-volume traffic environments. Arista Networks recommends using this feature judiciously.

Example

```
controller-1# show running-config managed-service Test
! managed-service
managed-service Test
service-interface switch CORE-SWITCH-1 ethernet13/1
1 dedup full-packet window 2
2 mask BIGSWITCH
3 slice l4-payload-start 0
!
4 netflow an-collector
collector 10.106.6.15 udp-port 2055 mtu 1500
```

This feature replaces the **`service-action`** command with sequential numbers. The allowed range of sequence numbers is ***1 -20000***. In the above example, the sequence numbering impacts the order in which the managed services influence the traffic.

Note: After upgrading to ***DANZ Monitoring Fabric (DMF) release 8.1.0*** and later, the service-action CLI is automatically replaced with sequence number(s).

Specific managed service statistics can be viewed via the following CLI command:

```
controller-1# show managed-service-device R740 stats Test
~~~
```

5.24 Sample Service

The Service Node forwards packets based on the **max-tokens** and **tokens-per-refresh** parameters using the DANZ Monitoring Fabric (DMF) Sample Service feature. The sample service uses one token to forward one packet.

After consuming all the initial tokens from the **max-tokens** bucket, the system drops subsequent packets until the `max-tokens` bucket refills using the **tokens-per-refresh** counter at a recurring predefined time interval of 10ms. Packet sizes do not affect this service.

Arista Networks recommends keeping the **tokens-per-refresh** value at or below **max-tokens**. For example, **max-tokens = 1000** and **tokens-per-refresh = 500**.

Setting the **max-tokens** value to 1000 means that the initial number of tokens is 1000, and the maximum number of tokens stored at any time is 1000.

The **max-tokens** bucket will be zero when the Service Node has forwarded 1000 packets before the first 10 ms period ends, leading to a situation where the Service Node is no longer forwarding packets. After every 10ms time interval, if the **tokens-per-refresh** value is set to 500, the **max-tokens** bucket is refilled using the **tokens-per-refresh** configured value, 500 tokens in this case, to pass packets the service tries to use immediately.

Suppose the traffic rate is higher than the refresh amount added. In that case, available tokens will eventually drop back to 0, and every 10ms, only 500 packets will be forwarded, with subsequent packets being dropped.

If the traffic rate is lower than the refresh amount added, a surplus of tokens will result in all packets passing. Since the system only consumes some of the tokens before the next refresh interval, available tokens will accumulate until they reach the **max-tokens** value of 1000. After 1000, the system does not store any surplus tokens above the **max-tokens** value.

To estimate the maximum possible packets passed per second (pps), use the calculation (1000ms/10ms) * **tokens-per-refresh** and assume the **max-tokens** value is larger than **tokens-per-refresh**. For example, if the **tokens-per-refresh** value is **5000**, then **500000** pps are passed.

The Sample Service feature can be used as a standalone Managed Service or chained with other Managed Services.

Use Cases and Compatibility

- Applies to Service Nodes
- Limit traffic to tools that cannot handle a large amount of traffic.
- Use the Sample Service before another managed service to decrease the load on that service.
- The Sample Service is applicable when needing only a portion of the total packets without specifically choosing which packets to forward.

5.24.1 Sample Service CLI Configuration

1. Create a managed service and enter the service interface.
2. Choose the sample managed service with the ***seq num*** **sample** command.
   a. There are two required configuration values: **max-tokens** and **tokens-per-refresh**. There are no default values, and the service requires both values.
   b. The **max-tokens** value is the maximum size of tokens in the token bucket. The service will start with the number of tokens specified when first configured. Each packet passed consumes one token. If no tokens remain, packet forwarding stops. Configure the **max-tokens** value from a range of ***1*** to the maximum uint64 (unsigned integer) value of ***9,223,372,036,854,775,807***.
   c. DMF refreshes the token bucket every 10 ms. The **tokens-per-refresh** value is the number of tokens added to the token bucket on each refresh. Each packet passed consumes one token, and when the number of tokens drops to zero, the system drops all subsequent packets until the next refresh. The number of tokens in the bucket cannot exceed the value of **max-tokens**. Configure the tokens-per-refresh value from a range of ***1*** to the maximum uint64 (unsigned integer) value of ***9,223,372,036,854,775,807***.

   The following example illustrates a typical Sample Service configuration

```
dmf-controller-1(config-managed-srv-sample)# show this
! managed-service
managed-service MS
!
3 sample
max-tokens 50000
tokens-per-refresh 20000
```

3. Add the managed service to the policy.

Show Commands

Use the **show running-config managed-service** ***sample_service_name*** command to view pertinent details. In this example, the `sample_service_name` is ***techpubs***.

```
DMF-SCALE-R450> show running-config managed-service techpubs

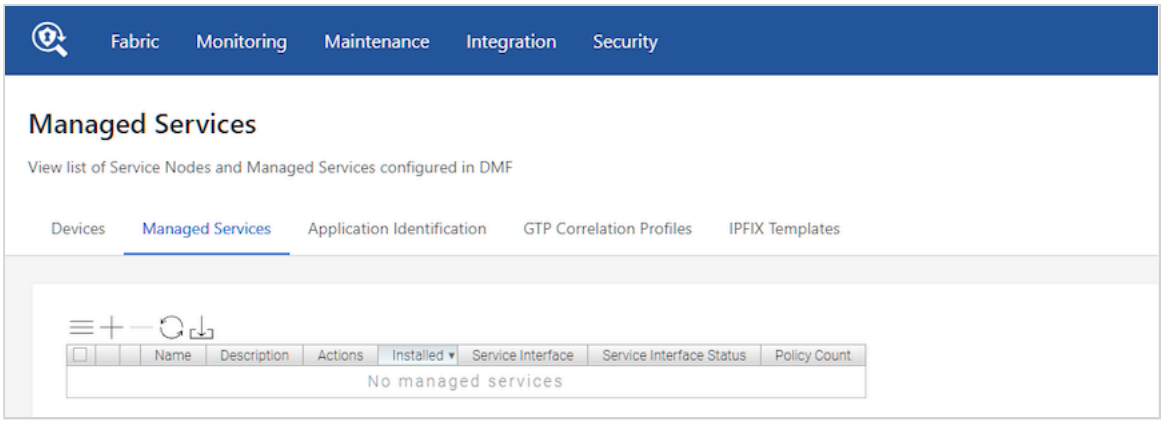
! managed-service
managed-service techpubs
  !
  1 sample
    max-tokens 1000
    tokens-per-refresh 500
DMF-SCALE-R450>
```

5.24.2 Sample Service GUI Configuration

Use the following steps to add a Sample Service.

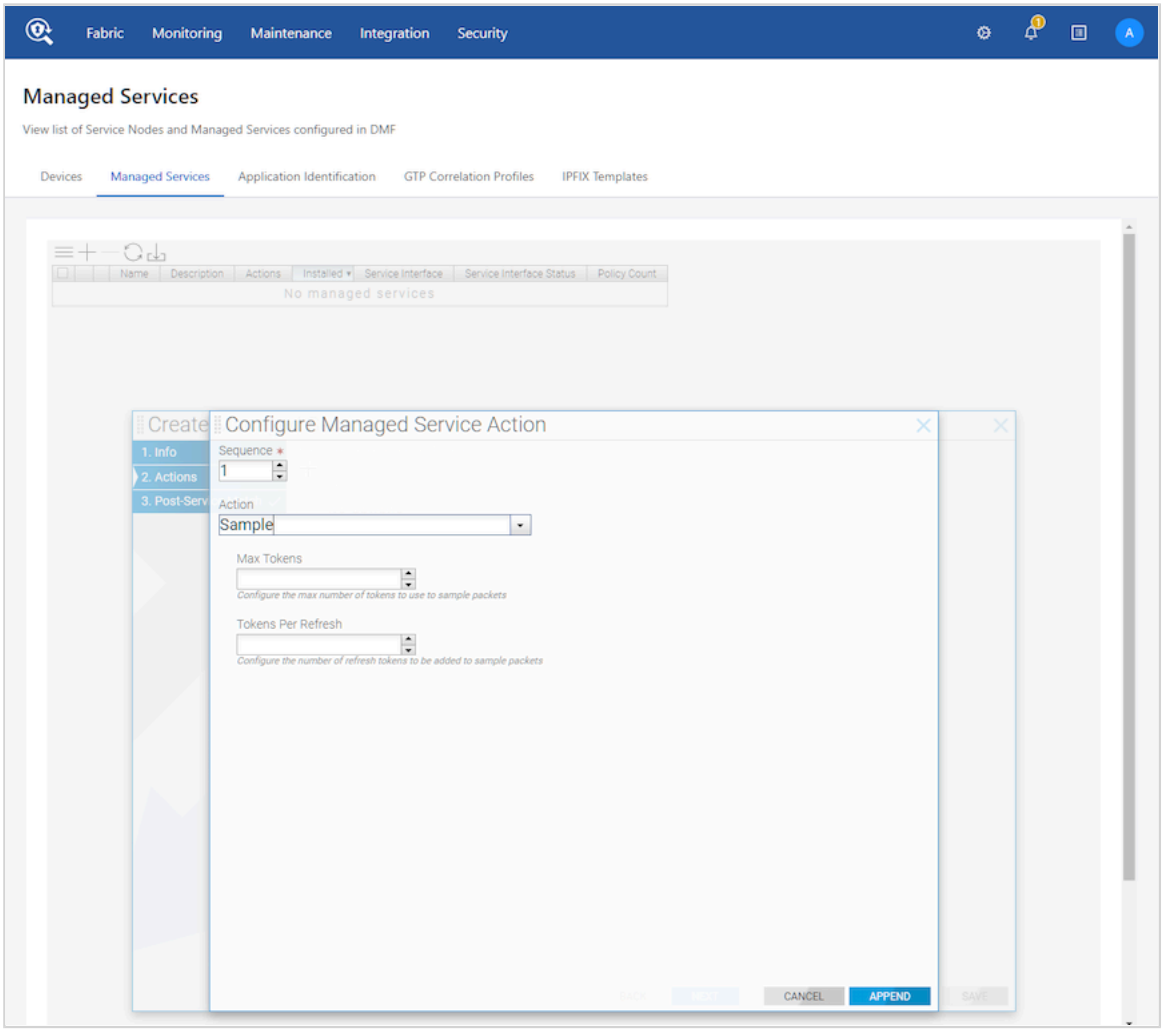
1. Navigate to the **Monitoring > Managed Services** page.

Figure 5-69: DMF Managed Services



2. Under the **Managed Services** section, select the **+** icon to create a new managed service. Go to the **Actions**, and select the **Sample** option in the **Action** drop-down. Enter values for **Max tokens** and **Tokens per refresh**.

Figure 5-70: Configure Managed Service Action



3. Select **Append** and then **Save**.

5.24.3 Troubleshooting Sample Service

Troubleshooting

- If the number of packets forwarded by the Service Node interfaces is few, the **max-tokens** and **tokens-per-refresh** values likely need to be higher.
- If fewer packets than the **tokens-per-refresh** value forward, ensure the **max-tokens** value is larger than the **tokens-per-refresh** value. The system discards any surplus refresh tokens above the **max-tokens** value.
- When all traffic forwards, the initial **max-tokens** value is too large, or the tokens refreshed by **tokens-per-refresh** are higher than the packet rate.
- When experiencing packet drops after the first 10ms post commencement of traffic, it may be due to a low **tokens-per-refresh** value. For example, calculate the minimum value of **max-tokens** and **tokens-per-refresh** that would lead to forwarding all packets.

Calculation Example

```
Traffic Rate : 400 Mbps
Packet Size - 64 bytes
400 Mbps = 400000000 bps
400000000 bps = 50000000 Bps
50000000 Bps = 595238 pps (Includes 20 bytes of inter packet gap in addition to the 64 bytes)
1000 ms = 595238 pps
1 ms = 595.238 pps
10 ms = 5952 pps
max-tokens : 5952 (the minimum value)
tokens-per-refresh : 5952 ( the minimum value)
```

5.24.4 Limitations

- In the current implementation, the Service Sample action is bursty. The token consumption rate is not configured to withhold tokens over time, so a large burst of incoming packets can immediately consume all the tokens in the bucket. There is currently no way to select what traffic is forwarded or dropped; it only depends on when the packets arrive concerning the refresh interval.
- Setting the **`max-tokens`** and **`tokens-per-refresh`** values too high will forward all packets. The maximum value is **`9,223,372,036,854,775,807`**, but Arista Networks recommends staying within the maximum values stated under the description section.

5.25 Flow Diff Latency and Drop Analysis

Latency and drop information help determine if there is a loss in a particular flow and where the loss occurred. A Service Node action configured as a DANZ Monitoring Fabric (DMF) managed service has multiple separate taps or spans in the production network and can measure the latency of a flow traversing through any pair of these points. It can also detect packet drops between any two points in the network if the packet only appears on one point within a specified time frame, currently set to 200ms.

Latency and drop analysis require Precision Time Protocol (PTP) time-stamped packets. The DMF PTP timestamping feature can do this as the packets enter the monitoring fabric, or the production network switches can also timestamp the packet.

The Service Node accumulates latency values by flow and sends IPFIX data records with each flow's 5-tuple and ingress and egress identifiers. It sends IPFIX data records to the Analytics Node after collecting a specified number of values for a flow or when a timeout occurs for the flow entry. The threshold count is 10,000 packets, and the flow timeout is 4 seconds.



Note: This feature is only supported in **`push-per-filter`** mode. Only basic statistics, such as **`min`**, **`max`**, and **`mean`**, are available. These statistics are the computed difference in timestamps, or latency, between two tap point pairs of packets within a flow.

Use the DMF Analytics Node to build custom dashboards to view and check the data.



Attention: The flow diff latency and drop analysis feature is switch dependent and requires PTP timestamping. It is supported on 7280R3 and 7800R3 switches.

5.25.1 Configure Flow Diff Latency and Drop Analysis Using the CLI

Configure this feature through the Controller as a Service Node action in managed services using the managed service action **`flow-diff`**.

Latency configuration configures multiple tap point pairs. DMF analyzes traffic flowing between these tap point pairs for latency or drops. A tap point pair comprises a source and a destination tap point, identified by the filter interface or filter interface group. Based on the latency configuration, configuring the Service Node with traffic metadata tells the Service Node where to look for tap point information, timestamps, and the IPFIX collector.

Configure appropriate DMF Policies to deliver traffic tapped from tap point pairs in the network to the configured Service Node interface for analysis.

Configuration Steps for `flow-diff`.

1. Create a managed service and enter the service interface.
2. Choose the **`flow-diff`** service action with the command: ***`seq num`*** **`flow-diff`**

   Note: The command will enter the **`flow-diff`** submode, which supports three configuration parameters: **collector**, **l3-delivery-interface**, and **`tap-point-pair`**. These all are required parameters.

3. Configure the IPFIX collector IP address by entering the following command: **`collector`** ***`ip-address`*** (the UDP port and MTU parameters are optional; the default values are **`4739`** and **`1500`**, respectively).
4. Configure the delivery interface by entering the command **`l3-delivery-interface`** ***`delivery interface name`***.
5. Configure the points for flow-diff and drop analysis using **`tap-point-pair`** parameters as specified in the following section. Multiple options to identify the tap-point include **`filter-interface`** and **`filter-interface-group`**. This command requires a source and a destination tap point.
6. Optional parameters are **`latency-table-size`**, **`sample-count-threshold`**, **`packet-timeout`**, and **`flow-timeout`**. The default values are **`large`**, **`10000`**, **`200`** ms, and **`4000`** ms, respectively.
7. The **`latency-table-size`** value determines the memory footprint of flow-diff action on the Service Node. This is an abstract concept managed entirely by the Service Node, and its meaning could evolve over time.
8. The **`sample-count-threshold`** value specifies the number of samples needed to generate a latency report. Every time a packet times out, it generates a sample for that flow. DMF generates a report if the flow reaches the sample threshold and resets the flow stats.
9. The **`packet-timeout`** value is the time interval in which timestamps are collected for a packet. It must be larger than the time it takes the same packet to appear at all tap points. Every timeout generates a sample for the flow associated with the packet.
10. The **`flow-timeout`** value is the time after which, when the flow no longer receives any packets, the system evicts the flow and generates a report. The timeout for a flow refreshes each time a new packet is received. If packets are continuously received for a flow below the flow timeout value, then the flow will never be evicted.

The following example illustrates configuring **flow-diff** using the steps mentioned earlier:

```
dmf-controller-1(config)# managed-service managed_service_1
dmf-controller-1(config-managed-srv)# service-interface switch delivery1 ethernet1
dmf-controller-1(config-managed-srv)# 1 flow-diff
dmf-controller-1(config-managed-srv-flow-diff)# collector 192.168.1.1
dmf-controller-1(config-managed-srv-flow-diff)# l3-delivery-interface l3-iface-1
dmf-controller-1(config-managed-srv-flow-diff)# tap-point-pair source filter-interface f1
 destination filter-interface f2
dmf-controller-1(config-managed-srv-flow-diff)# latency-table-size small|medium|large
dmf-controller-1(config-managed-srv-flow-diff)# packet-timeout 100
dmf-controller-1(config-managed-srv-flow-diff)# flow-timeout 4000
dmf-controller-1(config-managed-srv-flow-diff)# sample-count-threshold 10000
```

5.25.2 Configuring Tap Points

Configure tap points using **tap-point-pair** parameters in the **flow-diff** submode specifying two identifiers: **filter interface name**, and **filter-interface-group**.

```
dmf-controller-1(config-managed-srv-flow-diff)# tap-point-pair source <Tab>
filter-interface         filter-interface-group
```

The **filter-interface-group** option takes in any configured filter interface group used to represent a collection of tap points in **push-per-filter** mode. This is an optional command to use when a group of tap points exists, all expecting traffic from the same source or group of source tap points for ease of configuration. For example:

1. Instead of having two separate **tap-point-pairs** to represent A → B, A → C, use a **filter-interface-group G** = [B, C], and only one tap-point-pair A → G.

   ```
   dmf-controller-1(config-managed-srv-flow-diff# tap-point-pair source type A destination filter-
   interface-group G
   ```

2. With a topology like A → C and B → C, configure a **filter-interface-group G** = [A, B], and **tap-point-pair** G → C.

   ```
   dmf-controller-1(config-managed-srv-flow-diff # tap-point-pair source filter-interface-group G
    destination type C
   ```

There are some restrictions to keep in mind while configuring **tap-point-pairs**:

- A **source** and **destination** must exist and cannot refer to the same tap point
- You can only configure a maximum of 1024 tap points, therefore a maximum of 512 tap-point-pairs. This includes all managed services with **flow-diff** action, system-wide.
- The configured **filter-interface-group** must not overlap with other groups within the same managed service and cannot have more than 128 members.
- When configuring multiple **tap-point-pairs** using **filter-interface** and **filter-interface-group**, an interface part of a filter interface group cannot be used simultaneously as an individual source and destination within the same managed service.

If an interface group has more than one LAG grouping where each LAG contains multiple ids, only one member id of each LAG is required. Subsequent ids of the LAG will not have latency computed if they are found.

5.25.3 Configuring Policy

Configure the policies so that the same packet can be tapped from two independent points in the network and then sent to the Service Node.

After creating a policy, add the managed service with **flow-diff** action as shown below:

```
dmf-controller-1 (config-policy)# use-managed-service service name sequence 1
```

There are several things to consider while configuring policies in **push-per-filter** mode:

- Only one policy can contain the **flow-diff** service action.
- A policy should have all **filter-interfaces** and **filter-interface-groups** configured as tap points in the **flow-diff** configuration. Any missing filter interfaces and groups in a policy may result in reporting drops as these packets do not forward from one end of the **tap-point-pair** to the Service Node.
- It's also advisable not to add any filter interfaces (groups) that are not in the **tap-point-pairs** as their latency and drop analysis will not be done, causing unnecessary packets to be forwarded to the Service Node, which are then reported as **unexpected**.
- Ensure accurate timestamping and avoid unexpected report types by correctly ordering the source and destination for tap point pairs. Tap point pairs are not bidirectional so to compute latencies for both directions add the reverse ordering of the tap point pair.

Policies must have PTP timestamping enabled. To do so, use the following command:

```
dmf-controller-1 (config-policy)# use-timestamping
```

Configuring PTP Timestamping

This feature depends on configuring PTP timestamping for the packet stream going through the tap points. Refer to the Resources section for more information on setting up PTP timestamping functionality. Arista strongly recommends using **replace-src-mac** for this feature until revised, as performance and SN compatibility may vary for **add-header-after-l2**.

Analytics Node

IPFIX reports received on the analytics node can be of one of the following types, indicated by the **reportType** field:

- **0** is a valid latency report with the latency computed between the two tap point pairs.
- **1** means it is a drop report, which means the packet didn't arrive at its destination. If the egress identifier field is also **0**, the packet was only received at one tap point. If the egress identifier contains a nonzero value, that means one or a few of the LAG groupings associated with a destination interface group did not receive any packets associated with an id in that LAG. The egress identifier field is filled with the id of one of the members in that LAG group.
- **2** means unexpected, which can be caused by packets with identifiers not included in any tap point pairs or a misordering of timestamps where the destination comes before the source tap.

- **3** means overflow, where this same packet appears at too many tap points. Packets that surpass the maximum tap points are ignored.

VXLAN

When using VXLAN, the headers of these packets must be decapsulated before they can be processed by **flow-diff** on the service node. The filter switch must perform the decapsulation. If encapsulated packets appear at one tap point and non-encapsulated packets at the other endpoint, the service requires both of these packets to be decapsulated. Arista recommends using the following process to ensure **flow-diff** works correctly.

1. Configure the filter interface receiving the VXLAN encapsulated traffic to perform decapsulation.
2. If the receiving VXLAN traffic appears with non-default port number **4789**, configure the filter switch with a non-default port number as shown in the following example:

```
filter-switch-1# configure
filter-switch-1(config)# interface Vxlan 1
filter-switch-1(config-if-Vx1)# vxlan udp-port customer port

interface Vxlan1
   vxlan udp-port customer port
```

5.25.4 Show Commands

The following **show** commands provide helpful information.

The **show running-config managed-service *managed service*** command helps verify whether the **flow-diff** configuration is complete.

```
dmf-controller-1(config)# show running-config managed-service flow-diff
! managed-service
managed-service flow-diff
service-interface switch DCS-7050CX3-32S ethernet2/4
!
1 flow-diff
collector 192.168.1.1
l3-delivery-interface AN-Data
tap-point-pair source filter-interface ingress destination filter-interface egress
```

The **show managed-services *managed service*** command provides status information about the service.

```
dmf-controller-1(config)# show managed-services flow-diff
# Service Name Switch          Switch Interface Installed Max Post-Service BW Max Pre-Service BW Total Post-Service BW Total Pre-
Service BW
-|------------|---------------|----------------|---------|-------------------|------------------|---------------------|----------
----------|
1 flow-diff    DCS-7050CX3-32S ethernet2/4      True      25Gbps              25Gbps             624Kbps               432Mbps
```

The **show running-config policy *policy*** command checks whether the policy **flow-diff** service exists, whether **use-timestamping** is enabled, and the use of the correct filter interfaces.

```
dmf-controller-1(config)# show running-config policy p1
! policy
policy p1
  action forward
  filter-interface egress
  filter-interface ingress
  use-managed-service ms1 sequence 1
  use-timestamping
  1 match any
```

The **`show policy` *`policy`*** command provides detailed information about a policy and whether any errors are related to the **`flow-diff`** service. The **`Service Interfaces`** tab section shows the packets transmitted to the Service Node and IPFIX packets received from the Service Node.

```
dmf-controller-1 (config)# show policy flow-diff-1
Policy Name                             : flow-diff-1
Config Status                           : active - forward
Runtime Status                          : installed
Detailed Status                         : installed - installed to forward
Priority                                : 100
Overlap Priority                        : 0
# of switches with filter interfaces    : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces   : 1
# of filter interfaces                  : 1
# of delivery interfaces                : 1
# of core interfaces                    : 4
# of services                           : 1
# of pre service interfaces             : 1
# of post service interfaces            : 1
Push VLAN                               : 2
Post Match Filter Traffic               : 215Mbps
Total Delivery Rate                     : -
Total Pre Service Rate                  : 217Mbps
Total Post Service Rate                 : -
Overlapping Policies                    : none
Component Policies                      : none
Runtime Service Names                   : flow-diff
Installed Time                          : 2023-11-16 18:15:27 PST
Installed Duration                      : 19 minutes, 45 secs
~ Match Rules ~
# Rule
-|-----------|
1 1 match any
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF Switch   IF Name    State Dir Packets  Bytes       Pkt Rate Bit Rate Counter Reset Time
-|------|--------|----------|-----|---|--------|-----------|--------|--------|-------------------------------|
1 BP1    7280SR3E Ethernet25 up    rx  24319476 27484991953 23313    215Mbps  2023-11-16 18:18:18.837000 PST
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF  Switch    IF Name    State Dir Packets Bytes  Pkt Rate Bit Rate Counter Reset Time
-|-------|---------|----------|-----|---|-------|------|--------|--------|-------------------------------|
1 AN-Data 7050SX3-1 ethernet41 up    tx  81      117222 0        -        2023-11-16 18:18:18.837000 PST
~~~~~~~~~~~~~~~~~~~
```

5.25.5 Syslog Messages

The Flow Diff Latency and Drop Analysis feature does not create Syslog messages.

5.25.6 Troubleshooting

DMF Controller

Policies dictate how and what packets are directed to the Service Node. Policies must be able to stream packets from two distinct tap points so that the same packet gets delivered to the Service Node for **`flow-diff`** and **`drop`** analysis.

Possible reasons for flow-diff and drop analysis not working are:

- In **`push-per-filter`** mode, the policy bound to managed service with flow-diff action is missing filter interfaces or groups that constitute a **`tap-point-pair`**.

A policy programmed to use managed service with **`flow-diff`** action can fail for several reasons:

- The L3 delivery interface or Collector configuration is missing.
- The **`tap-point-pair`** configuration is incomplete:
  - The **`source`** or **`destination`** tap **`points`** are missing.
  - Using a **`policy-name`** identifier in **`push-per-filter`** mode.
- There are more than 512 **`tap-point-pairs`** configured or more than 1024 distinct tap points. These limits are global across all flow-diff managed services.
- **`filter-interface-groups`** overlap with each other within the same managed service or have more than 128 group members.
- **`filter-interface`** is being used individually as a tap point and as a part of some **`filter-interface-group`** within the same managed service.

Reasons for failure are available in the runtime state of the policy and viewed using the **`show policy`** ***`policy name`*** command.

A lack of computed latency reports can mean two things:

- Not reaching the **`sample-count-threshold`** value. Either lower the **`sample-count-threshold`** value until reports are generated or increase the amount of unique packets per flow.
- Flows may not be evicted. The time specified in **`flow-timeout`** will refresh every time a new packet is received on that flow before the timeout occurs. If a flow continuously receives packets, it will never time out. Lower the **`flow-timeout`** value if the default of **`4`** seconds causes flows not to expire. Take this action if the sample threshold counter is taking a long time due to low traffic volume for that flow. Adjust the **`sample-count-threshold`** value before attempting **`flow-timeout`** changes. Once a flow expires, the system generates a report for that flow.

For **`packet-timeout`**, this value must be larger than the time expected to receive the same packet on every tap-point. For A->B, if it takes 20ms for the packet to appear at B, the **`packet-timeout`** must be larger than this time frame to collect timestamps for both these tap points and compute a latency in one sample.



Note: When using a rate limit, the **packet-timeout** value may need to be increased.

If all the reports are type 2 **unexpected**, ensure the taps are in the correct order as configured on the Controller. Also, check the topology is correct and that there are no unexpected taps. If there are only a few **unexpected** reports, it is most likely an issue with the switch ordering the timestamps. You can ignore these or add the reverse direction onto the topology to turn these into type **0** reports. If there are many drop reports, confirm that the same packet is received at all relevant taps within the packet-timeout window, which defaults to 200ms.

5.25.7 Limitations

- This feature is only supported in **`push-per-filter`** mode. Any config in **`push-per-policy`** mode will not work as intended.
- A maximum of **`512 tap-point-pairs`** and **`1024`** distinct tap points are allowed.
- Only 40 timestamped instances of a packet are allowed.
- The **`filter-interface-group`** used as a tap point must not overlap with any other group within the same managed service and must not have more than 128 members.
- A filter interface cannot be used individually as a tap point and as a part of a **`filter-interface-group`** simultaneously within the same managed service.
- There is no chaining if a packet flows through three or more tap points in an A->B->C->...->Z topology. The only computed latency reports is for `tap-point-pairs` A->B, A->C, …, and A->Z if these links are specified, but B->C, C->D, etc., will not be computed.
- Service node interfaces can have multiple lcores depending on the SKU. An lcore is a logical core which processes traffic on an interface. Hardware RSS firmware in some Service Node SKU currently cannot parse L2 header timestamps, so all packets are sent to the same lcore; however, RSS does distribute packets correctly to multiple lcores when using src-mac timestamping.
- Each packet from the L3 header and onwards gets hashed to a 64-bit value; if two packets hash to the same value, assume the underlying packets are the same.
- Currently, on the **`flow-diff`** action in the Service Node, if packets are duplicated so that N copies of the same packet are received:
  - N-1 latencies are computed.
  - The ingress identifier is the earliest timestamp.
- The system reports timestamps as unsigned 32-bit values, with the maximum timestamp being **`2^32-1`**, corresponding to approximately 4.29 seconds.
- Only min/mean/max latencies are currently reported.
- If there are switch timestamping issues, then these statistics may have high outliers.
- Synchronize the time between switches for this feature to work properly, or latency calculation may be inaccurate.
- Packets are hashed from the L3 header onwards, meaning if there is any corrupted data past the L3 header, it will lead to drop reports. The same packet must appear at two tap points to generate a computed latency report.
- In A->B, if B packets appear before A, an **`unexpected`** type report is generated.
- At whichever tap point the packet first appears with the earliest timestamp, it is considered the source.
- While switching the latency configuration, the system may generate a couple of unexpected or drop reports at the beginning.
- A LAG may affect RSS on the service node which may affect performance.

- Users must have good knowledge of network topology when setting up tap points and configuring timeout or sample threshold values. Improper configuration may lead to drop or unexpected reports.
- Occasionally switch timestamping causes the system to generate a few type **2 unexpected** reports.
- The system may generate drop reports from certain protocols, such as OSPF.
- The amount of packets per second and packet size influences performance since packets are individually hashed. The amount of tap point pairs and interface groups also affects performance.

5.25.8 Resources

- PTP Timestamping (https://www.arista.com/en/support/toi/dmf-8-4-0/18066-ptp-timestamping-on-7280-switches)

5.26 Sharing L3 Delivery Interfaces Across Services

Overview

A DMF Controller will allow multiple managed services to share a delivery interface with an IP address, commonly called an L3 delivery interface. These interfaces redirect the packets processed by managed services to the required tool nodes for further analysis. Sharing an L3 delivery interface is useful when applying different actions to a packet that otherwise cannot be chained together in one managed service when sending it to the same destination.

Currently, only a few managed service actions require L3 delivery interfaces. Since these actions must be the terminating action in a given service action chain, DMF allows different services to use the same L3 delivery interface.

This feature applies to all platforms that support managed services with L3 delivery interfaces and the following managed service actions using L3 deliveries:

- Netflow
- IPFIX
- UDP Replicate
- TCP Analysis
- App ID
- Flow Diff



Note: DMF 8.8 supports multiple L3 delivery interfaces on the same subnet and gateway. Refer to the **`show running-config`** command example in the Show Commands section.

Configuration

Log into the DMF Controller, select a switch, and create an L3 delivery interface.

```
dmf-controller-1> enable
dmf-controller-1# configure
dmf-controller-1(config)# switch switch1
dmf-controller-1(config-switch)# interface ethernet31
dmf-controller-1(config-switch-if)# role delivery interface-name l3-d1 ip-address 0.0.0.1 nexthop-
ip 0.0.0.1 255.255.255.0 nexthop-arp-interval 5
dmf-controller-1(config-switch-if)# rate-limit 1000
```

Create two or more managed services with the actions described previously as the last entry in the sequence. Use the **`l3-d1`** interface created earlier as the **`l3-delivery-interface`** for these services. The configuration commands needed may be different for each action. For example, using a managed service with **`netflow`** action and another with the **`udp-replicate`** action as shown in the following command sequences:

```
dmf-controller-1> enable
dmf-controller-1# configure
```

```
dmf-controller-1(config)# managed-service ms1
dmf-controller-1(config-managed-srv)# service-interface switch switch1 ethernet1
dmf-controller-1(config-managed-srv)# 1 netflow
dmf-controller-1(config-managed-srv-netflow)# collector 0.0.0.2 udp-port 2055 mtu 1500
dmf-controller-1(config-managed-srv-netflow)# l3-delivery-interface l3-d1
```

Show Commands

Use the **`show running-config switch switch1 interface`** ***`l3 interface`*** command to view the configured L3 interface:

```
dmf-controller-1> show running-config switch switch1 interface ethernet31

! switch
switch switch1
  !
  interface ethernet31
    force-link-up
    rate-limit 1000
    role delivery interface-name l3-d1 ip-address 0.0.0.1 nexthop-ip 0.0.0.2 255.255.255.0
 nexthop-arp-interval 5
```

Use the **`show running-config`** command to view L3 interfaces using the same gateway:

```
dmf-controller-1>(config-switch-if)# show running-config switch
! switch
switch core1
mac 52:54:00:b7:c5:c4
admin hashed-password $6$92Yy8I4C$dWgXuHgY3qD9PNmvFHQoCQ0VlLhTR43Hr5jEAoO7nA.6di3NoIgme
gdGAQe3bJ4h55KQ6yDGzFIXgQER0yxuQ1
!
interface ethernet10
role delivery interface-name TOOL1 ip-address 192.168.0.10 nexthop-ip 192.168.0.100 255.255.255.0
!
interface ethernet11
role delivery interface-name TOOL2 ip-address 192.168.0.11 nexthop-ip 192.168.0.100 255.255.255.0
!
interface ethernet20
role delivery interface-name TOOL3 ip-address 192.168.0.20 nexthop-ip 192.168.0.100 255.255.255.0
```

Use the **`show running-config managed-service`** command to view the managed service running config:

```
dmf-controller-1> show running-config managed-service

! managed-service
managed-service ms1
  service-interface switch switch1 ethernet1
  !
  1 netflow
    collector 0.0.0.2 udp-port 2055 mtu 1500
    l3-delivery-interface l3-d1

managed-service ms2
  service-interface switch switch1 ethernet2
  !
  1 udp-replicate l3-d1
    in-dst-ip 1.1.1.1
    out-dst-ip 2.2.2.2
    Out-dst-ip 1.1.1.1
```

Use the **`show switch switch1 interface`** ***`l3 interface`*** command to view the runtime status of the interface:

```
dmf-controller-1> show switch switch1 interface ethernet31
```

```
# IF Name    MAC Address                Config State Adv. Features Curr Features Supported
 Features
-|----------|--------------------------|------|-----|-------------|-------------|--------------
----|
1 ethernet31 5c:16:c7:14:46:bb (Arista) up     up    10g           10g           10g
```

Use the **`show managed-service`** command to view the runtime status of the managed services:

```
dmf-controller-1> show managed-service
~~~
```

5.27 Service Node Management Migration L3ZTN

After the first boot (initial configuration) is completed, in the case of Layer-3 topology mode an administrator can move a Service Node (SN) from an old DANZ Monitoring Fabric (DMF) Controller to a new one via the CLI.



Note: For appliances to connect to the Controller in Layer-3 Zero Touch Network (L3ZTN) mode, you must configure the Controller's deployment mode as `pre-configure`.

To migrate a Service Node's management to a new Controller, follow the steps outlined below:

1. Remove the Service Node from the old Controller using the following command:

```
controller-1(config)# no service-node service-node-1
```

2. Connect the data NICs' `sni` interfaces to the new core fabric switch ports.
3. SSH to the Service Node and configure the new Controller's IP address using the **`zerotouch l3ztn controller-ip`** command:

```
service-node-1(config)# zerotouch l3ztn controller-ip 10.2.0.151
```

4. Get the management MAC address (of interface `bond0`) of the Service Node using the following command:

```
service-node-1(config)# show local-node interfaces
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Interfaces  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Interface Master Hardware address         Permanent hardware address Operstate Carrier Bond mode     Bond role
---------|------|------------------------|--------------------------|---------|-------|-------------|---------|
bond0            78:ac:44:94:2b:b6 (Dell)                            up        up      active-backup
```

5. Add the Service Node and its bond0 interface's MAC address (obtained in the step above) to the new Controller:

```
controller-2(config)# service-node service-node-1
controller-2(config-service-node)# mac 78:ac:44:94:2b:b6
```

6. After associating the Service Node with the new Controller, reboot the Service Node.
7. Once the Service Node is back online, the Controller should receive a ZTN request. If the Service Node's image differs from the Service Node image file on the new Controller, the mismatch triggers the Service Node to perform an auto-upgrade of the image and to reboot twice.

```
controller-2# show zerotouch request
41  78:ac:44:94:2b:b6 (Dell)    10.240.156.10 get-manifest    2024-06-12 23:14:42.284000 UTC ok    The request has succeeded
56  24:6e:96:78:58:b4 (Dell)    10.240.156.91 get-manifest    2024-06-12 23:13:38.633000 UTC ok    The request has succeeded
```

8. Then the Service Node should appear as a member of the new DMF fabric, which you can verify by using the following command:

```
controller-2# show service-node service-node-1 details
```

Chapter 6

DMF Recorder Node

This chapter describes configuring the DANZ Monitoring Fabric (DMF) Recorder Node (RN) to record packets from DMF filter interfaces. For related information, refer to the following:

- Installing and configuring DMF Recorder Node: ***DANZ Monitoring Fabric Deployment Guide***
- Integrating DMF Recorder Node with Analytics: ***Arista Analytics User Guide***
- Stenographer Queries: Stenographer Reference for DMF Recorder Node
- DMF Recorder Node REST API: DMF Recorder Node REST APIs

6.1 Overview

The DANZ Monitoring Fabric (DMF) Recorder Node (RN) integrates with the DMF for single-pane-of-glass monitoring. A single DMF Controller can manage multiple RNs, delivering packets for recording through Out-of-Band policies. The DMF Controller also provides central APIs for packet queries across one or multiple RNs and for viewing errors, warnings, statistics, and the status of connected RNs.

A DMF out-of-band policy directs matching packets for recording to one or more RNs. An RN interface identifies the switch and port used to attach the RN to the fabric. A DMF policy treats these as delivery interfaces and adds them to the policy so that flows matching the policy are delivered to the specified RN interfaces.

6.2 Configuration Summary

At a high level, perform the following steps to configure the Recorder Node (RN).

Step 1: Add the RN instance to the DMF Controller.

Step 2: Define DMF policy to select the traffic to forward to the RN.

Step 3: View and analyze the recorded traffic.

Refer to the Recorder Node User Interface section and the *Arista Analytics User Guide* for more information.

6.3 Indexing Configuration

The Recorder Node (RN) indexing configuration defines the fields used to query packets on the RN. By default, DMF enables all indexing fields in the indexing configuration. Selectively disable the specific indexing fields not required in RN queries.

Disabling indexing fields has two advantages. First, it reduces the index space required for each packet recorded. Second, it improves query performance by reducing unnecessary overhead. Arista recommends disabling unnecessary indexing fields.

The RN supports the following indexing fields:

- MAC Source
- MAC Destination
- VLAN 1: Outer VLAN ID
- VLAN 2: Inner/Middle VLAN ID
- VLAN 3: Innermost VLAN ID
- IPv4 Source
- IPv4 Destination
- IPv6 Source
- IPv6 Destination
- IP protocol
- Port Source
- Port Destination
- MPLS
- Community ID
- MetaWatch Device ID
- MetaWatch Port ID



Note: Enable the Outer VLAN ID indexing field to query the RN using a DANZ Monitoring Fabric (DMF) policy name or a DMF filter interface name.

To understand leveraging an indexing configuration, consider the following examples:

Example 1: To query packets based on applications defined by unique transport ports, disable all indexing fields except source and destination transport ports, saving only transport ports as metadata for each packet recorded. This technique greatly reduces per-packet index space consumption and increases RN query speed.

However, this will impact an effective query on any other indexing field because that metadata was not saved when the packets were recorded.

Example 2: The RN supports community ID indexing, a hash of IP addresses, IP protocol, and transport ports that identify a flow of interest. Suppose the RN use case is to query based on community ID. In that case, indexing on IPv4 source and destination addresses, IPv6 source and destination addresses, IP protocol, and transport port source and destination addresses might be redundant.

6.4 Pre-buffer Configuration and Events

The Recorder Node (RN) pre-buffer is a circular buffer recording received packets. When enabled, the pre-buffer feature allows for the retention of the packets received by the RN for a specified length of time prior to an event that triggers the recording of buffered and future packets to disk. Without an event, the RN will record into this buffer, deleting the oldest packets when the buffer reaches capacity.

When an RN event is triggered, DMF saves packets in the pre-buffer to disk. The packets received from the time of the event trigger to the time of the event termination are saved directly to disk upon termination of the event. However, the received packets are also retained in the pre-buffer until the next event is triggered. By default, the pre-buffer feature is disabled, indicated by a value of zero minutes.

For example, when configuring the pre-buffer to thirty minutes, the buffer will receive up to thirty minutes of packets. When triggering an event, DMF records the packets currently in the buffer to disk, and packets newly received by the RN bypass the buffer and are written directly to disk until the termination of the event. When terminating the event, the pre-buffer resets, accumulating received packets for up to the defined thirty-minute pre-buffer size.

The packets affiliated with an event can be queried, replayed, or analyzed using any RN query. Each triggered event is identified by a unique, user-supplied name, used in the query to reference packets recorded in the pre-buffer before and during the event.

6.5 Using an Authentication Token

When using a DANZ Monitoring Fabric (DMF) Controller authentication token, the Recorder Node (RN) treats the DMF Controller as an ordinary client, requiring it to present valid credentials either in the form of an HTTP basic username and password or an authentication token.

Static authentication tokens are pushed to each RN as an alternative form of authentication in headless mode when the DMF Controller is unreachable or by third-party applications that do not have or do not need Controller credentials.

6.6 Recorder Node User Interface

Overview

DANZ Monitoring Fabric (DMF) 8.7.0 introduces a redesigned **Recorder Node** (RN) UI with an improved configuration workflow, monitoring page, and query features.

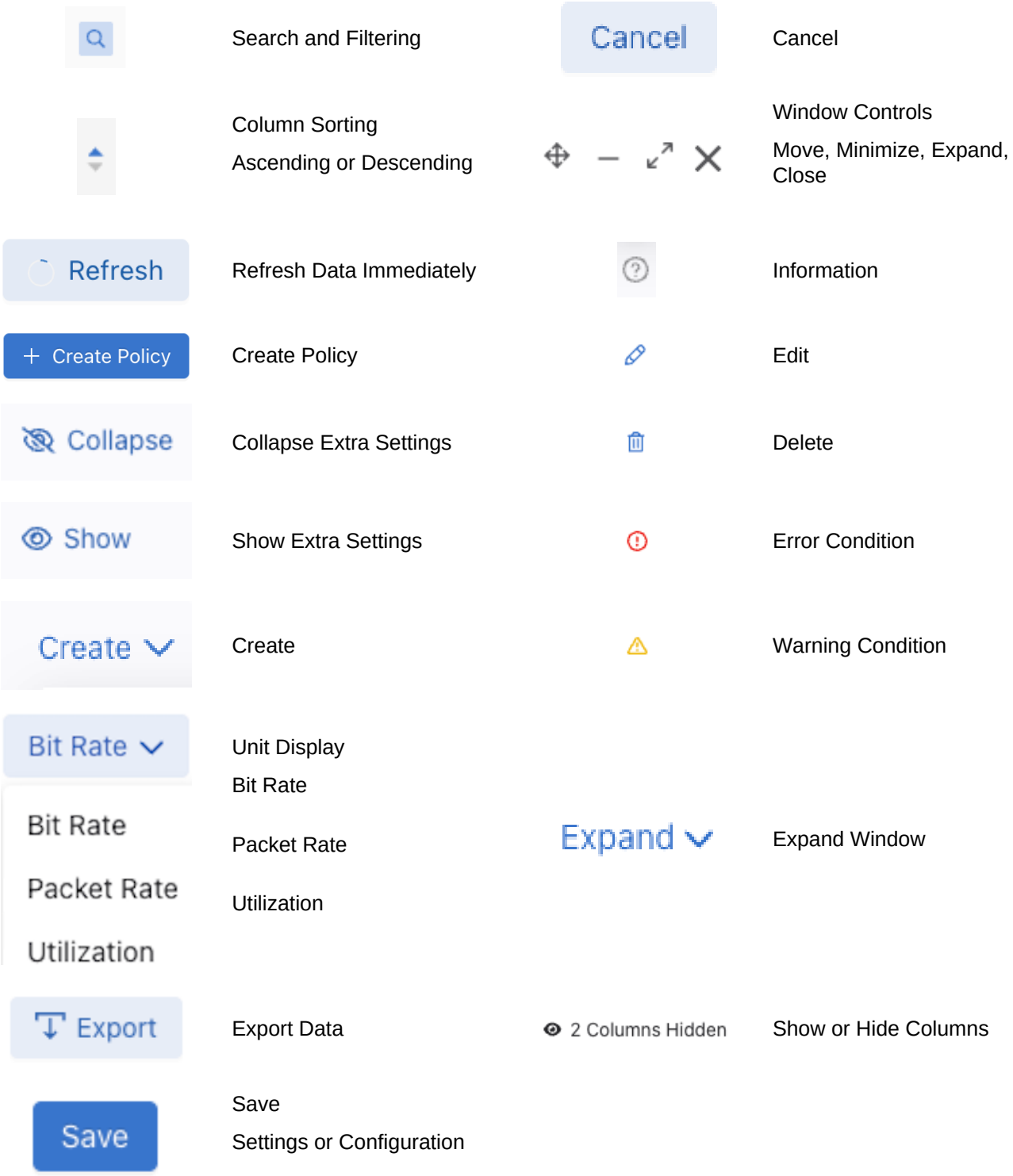
Common Features and Functions

Recorder Node Dashboard Layout

6.6.1 Features and Functions

There are several functions used in the DMF Recorder Node that are used throughout the user interface (UI).

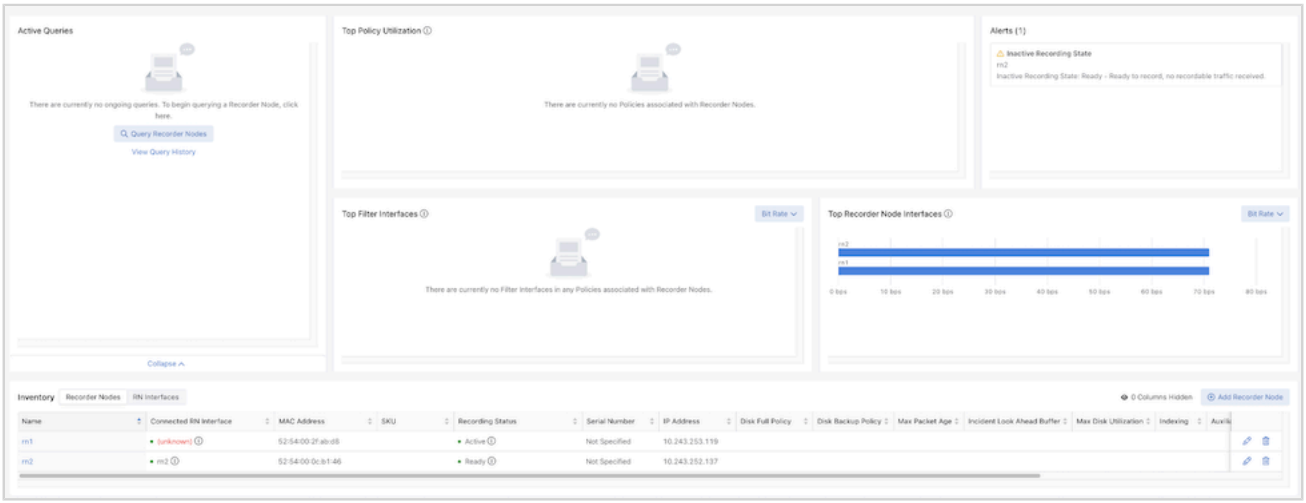
These include:



Recorder Node Dashboard Layout

Active Queries: Expand the Active Queries widget to view ongoing queries.

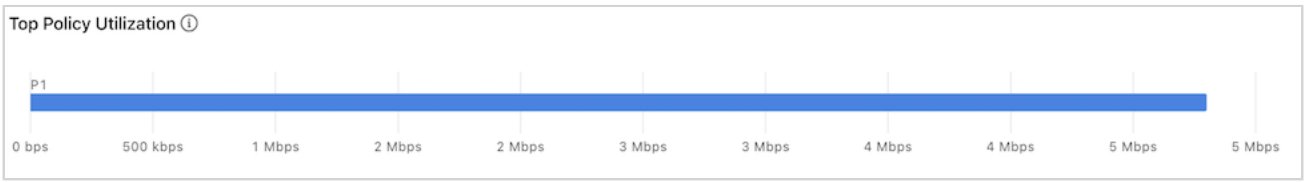
Figure 6-1: Active Queries



Note: Active Queries fetch queries periodically; if a query completes quickly, the query may not appear in the widget.

Top Policy Utilization: The chart displays up to 5 top policies associating the Recorder Node interfaces with their respective bit rates.

Figure 6-2: Top Policy Utilization



Alerts: The **Alerts** widget displays Recorder Node warnings or errors. Any errors appear in the **Alerts** drop-down in the menu bar.

Figure 6-3: Alert Messages

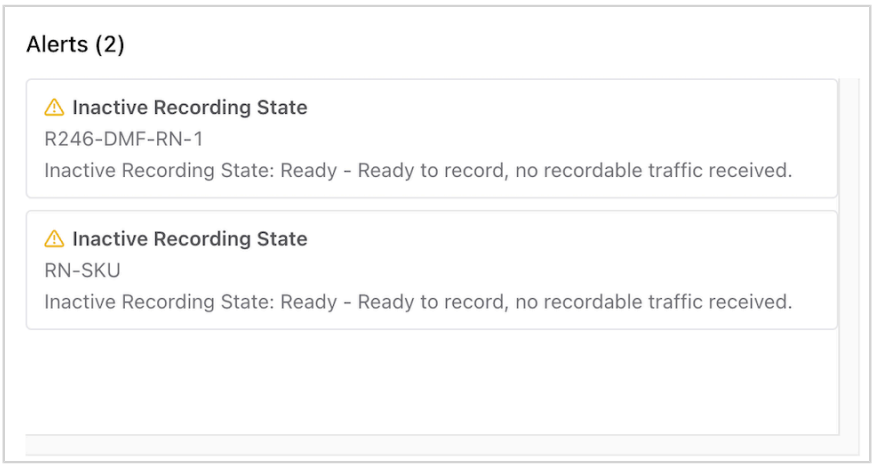
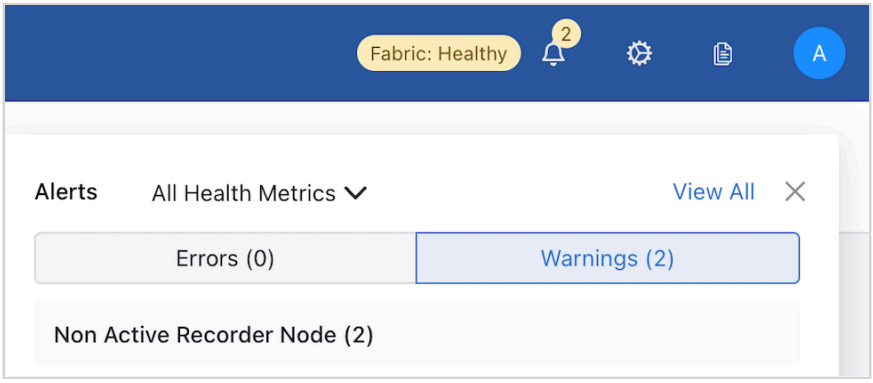


Figure 6-4: Fabric Health Status



Top Filter Interfaces: **Top Filter Interfaces** displays the **Top Filter Interfaces** (up to five) attached to a policy with associated **Recorder Node** interfaces. Select the **Unit** drop-down to update the data with the

selected unit type. The selection persists until changed. Hovering over a bar displays the throughput for the interface.

Figure 6-5: Top Filter Interfaces

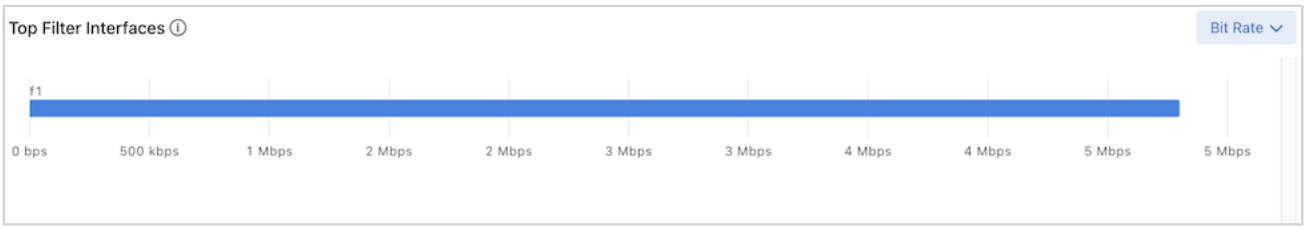
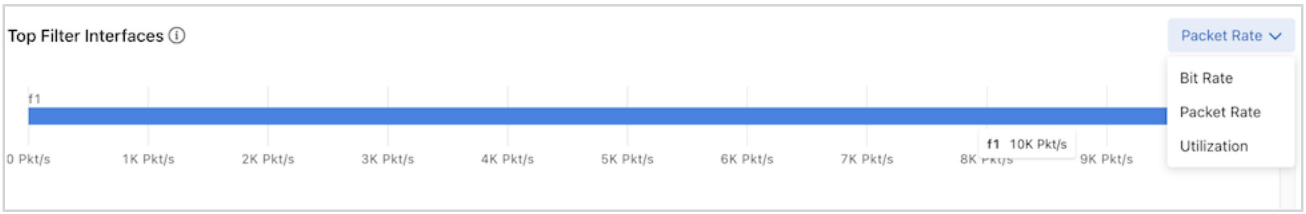
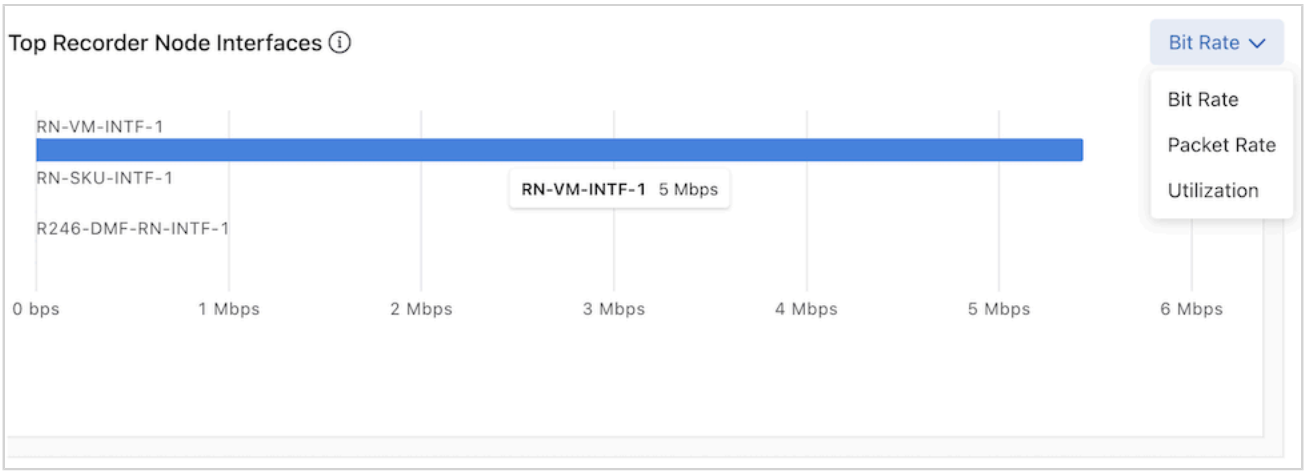


Figure 6-6: Top Filter Interfaces - Units



Top Recorder Node Interfaces: **Top Recorder Node Interfaces** displays the **Top Recorder Node Interfaces** (up to five) attached to a policy with associated **Recorder Node** interfaces. Select the **Unit** drop-down to update the data with the selected unit type. The selection persists until changed. Hovering over a bar displays the throughput for the interface.

Figure 6-7: Top Recorder Node Interfaces



Inventory Table:

The **Inventory Table** is located in the lower section of the **Recorder Nodes** dashboard and displays all **Recorder Node** instances configured on the DMF Fabric. Selecting **RN Interfaces** displays all **Recorder Node Interfaces** configured in DMF.

Note: Even if a **Recorder Node Interface** is not configured, but the fabric detects a **Recorder Node**, it will show a partial entry in the **Recorder Node Interfaces** table.

Figure 6-8: Inventory - Recorder Nodes

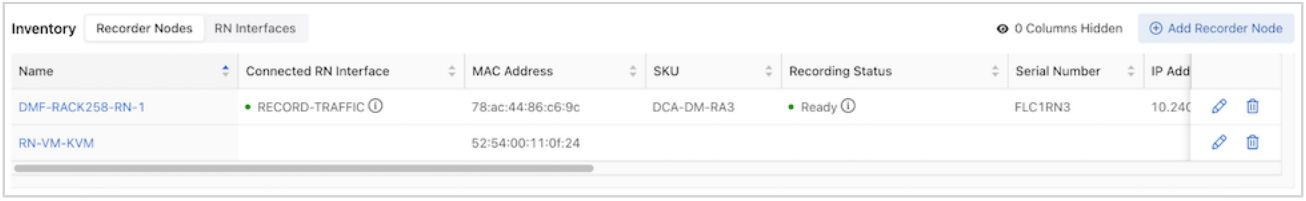
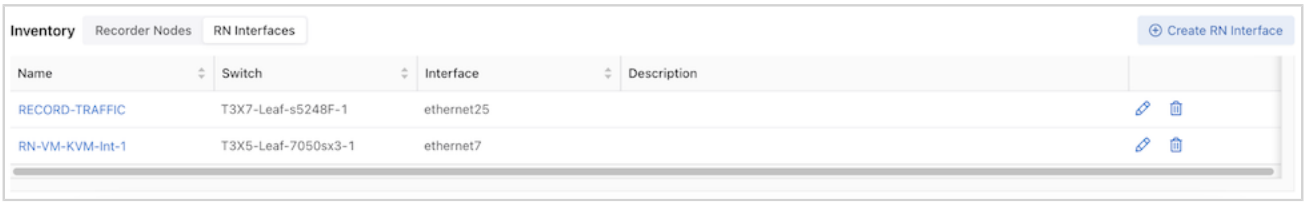


Figure 6-9: Inventory - Recorder Node Interfaces



Select **Edit** to modify the **Edit Recorder Node Interface** or **Edit Recorder Node** configuration.

Figure 6-10: Edit Recorder Node Interface

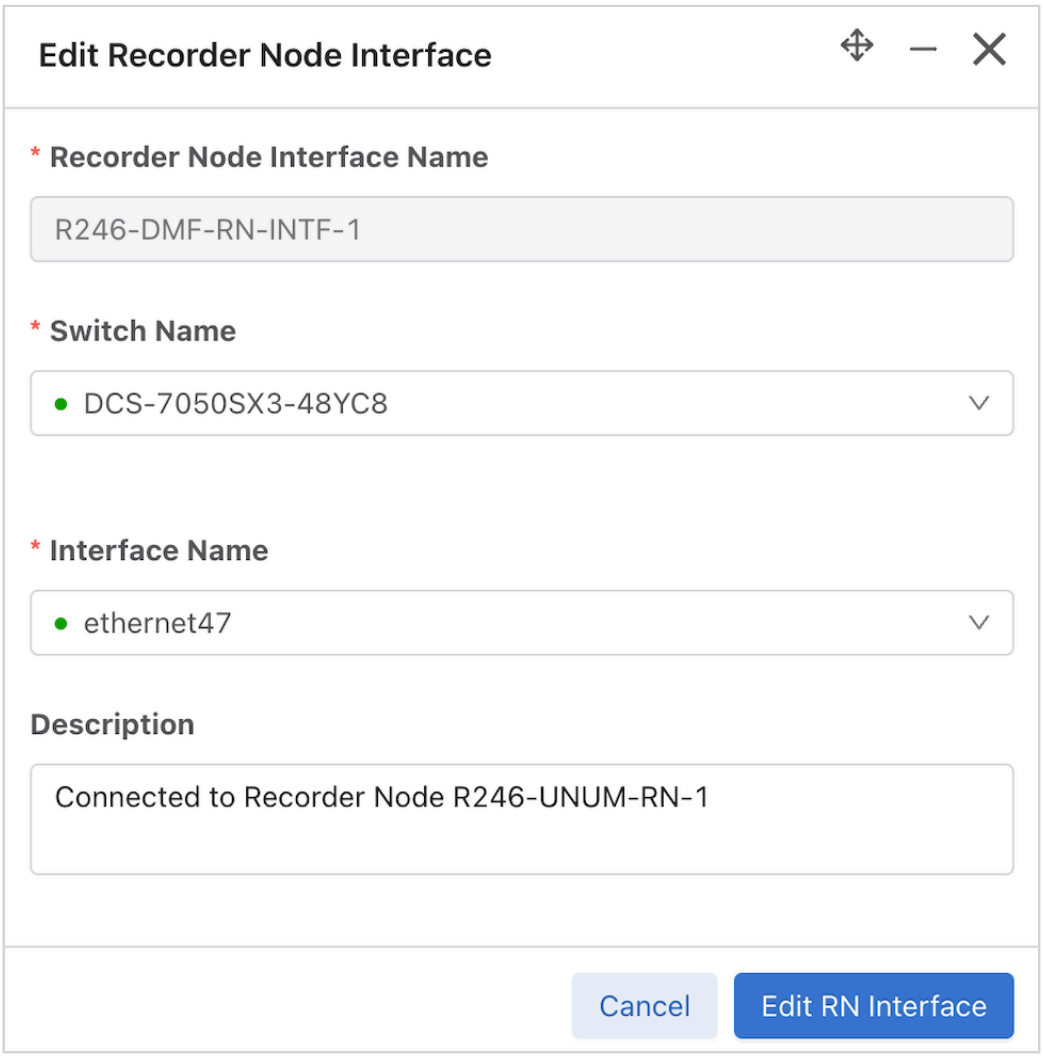
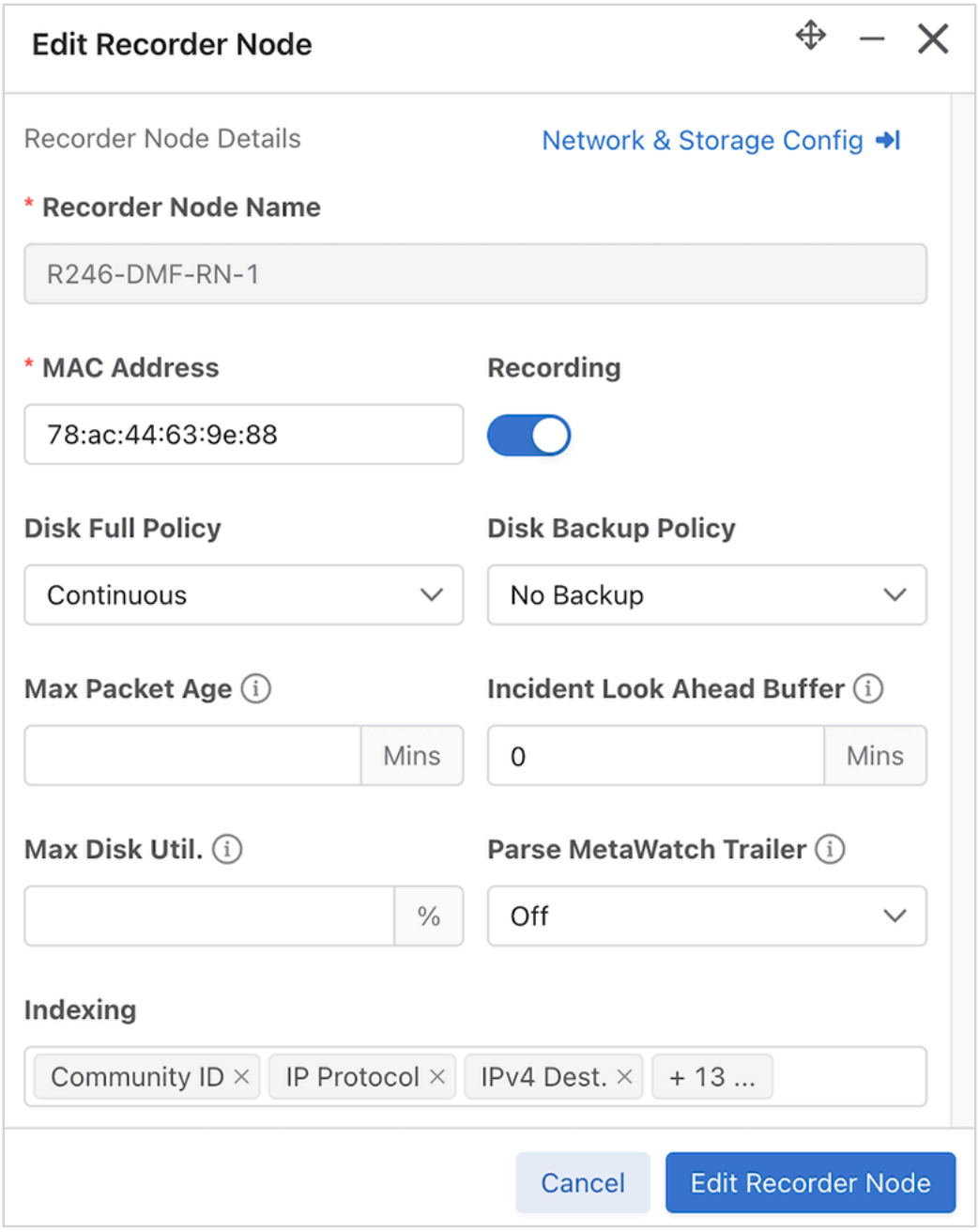


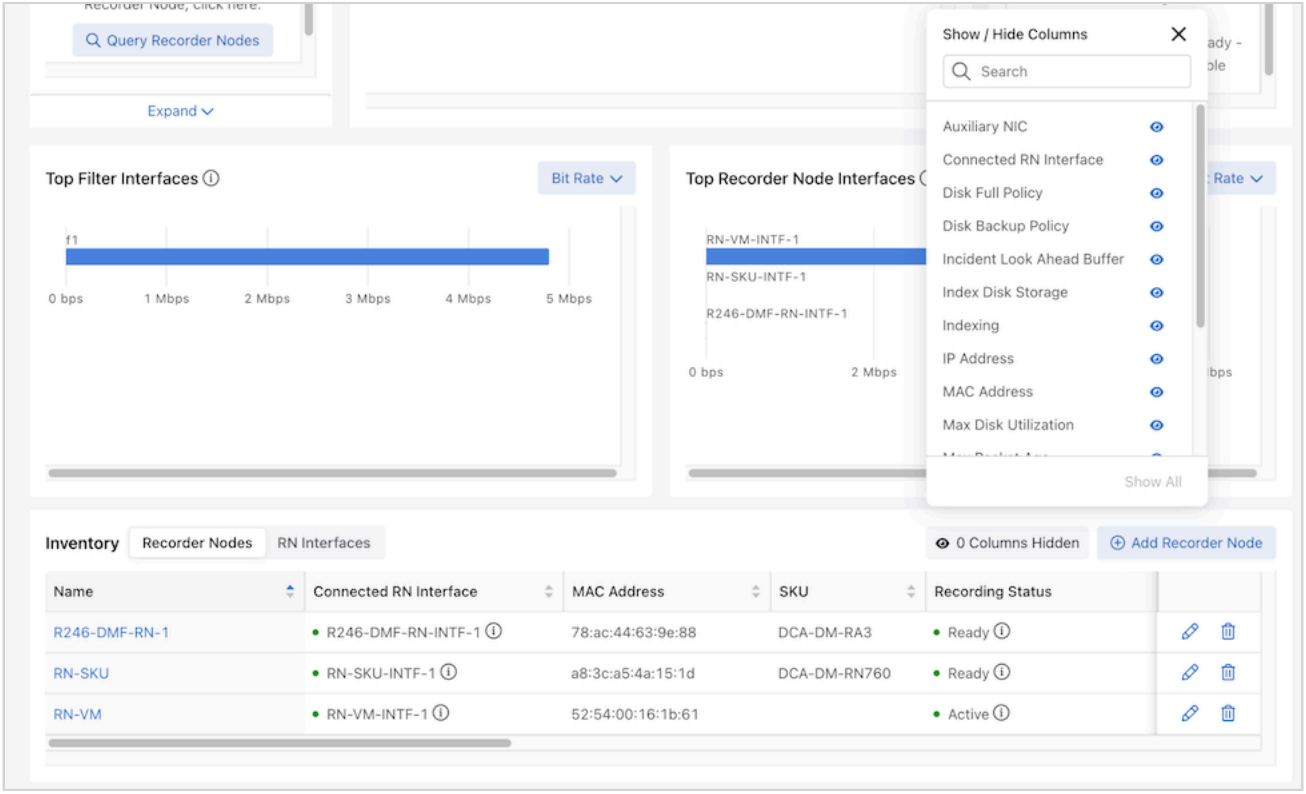
Figure 6-11: Edit Recorder Node



Show / Hide Columns determine the information displayed in the dashboard and are user-selectable. Selections persist until changed.

Note: The Name column cannot be hidden.

Figure 6-12: Show / Hide Columns

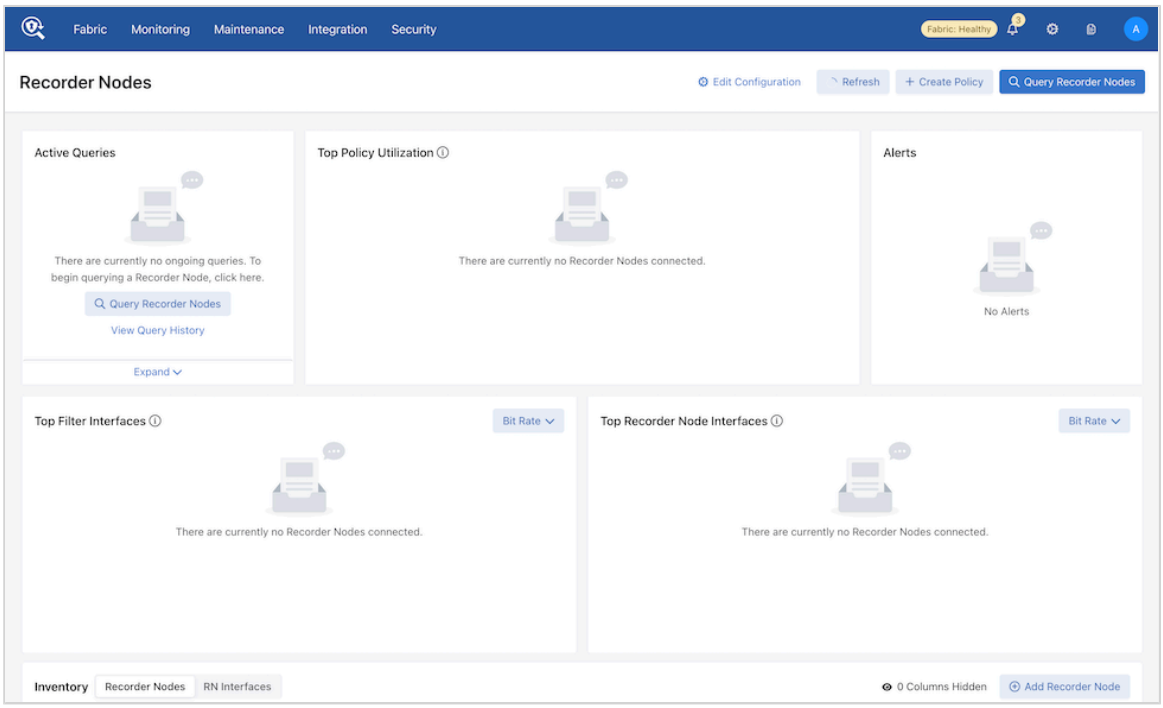


6.6.2 Onboard a Recorder Node

To configure a Recorder Node (RN) or update the configuration of an existing RN, perform the following steps:

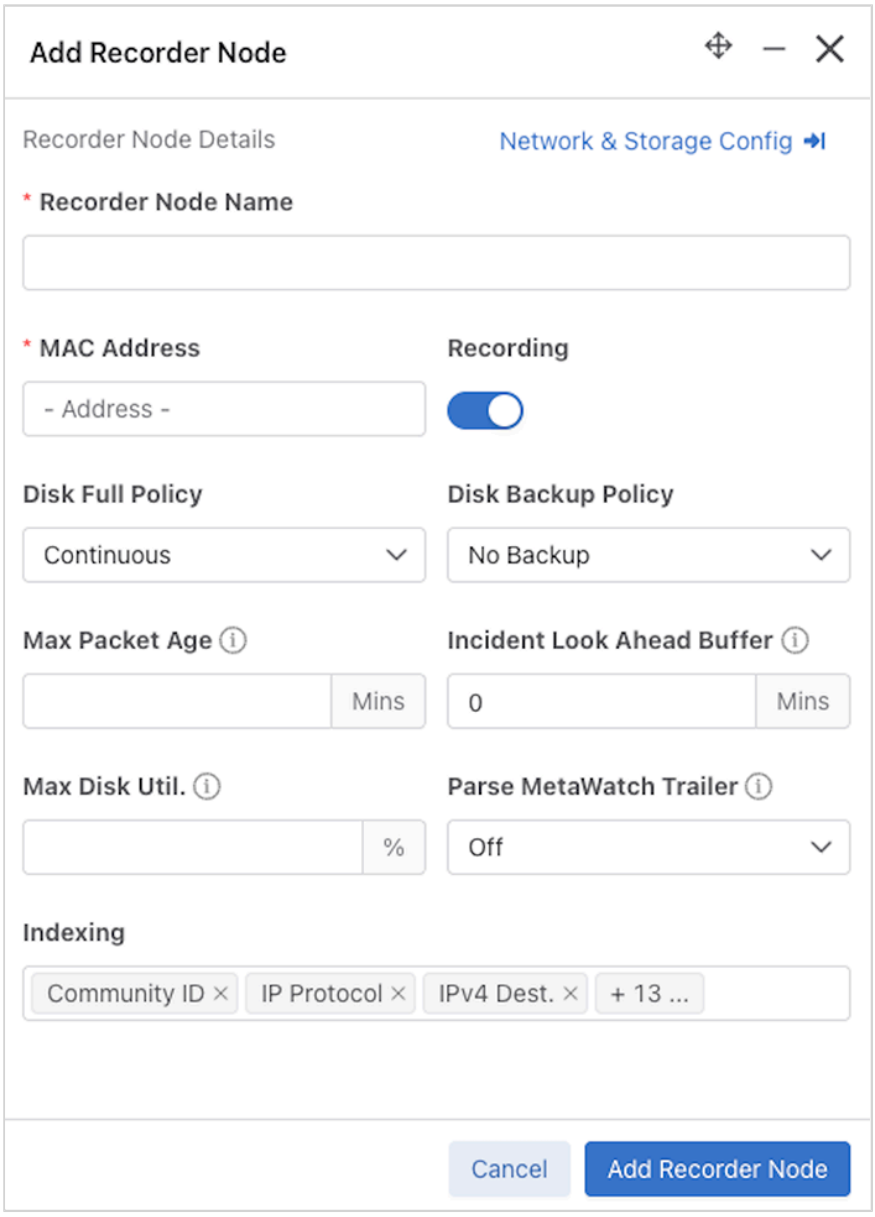
1. Select **Monitoring** > **Recorder Nodes** from the main menu bar of the DANZ Monitoring Fabric (DMF) GUI.

   Figure 6-13: Recorder Nodes



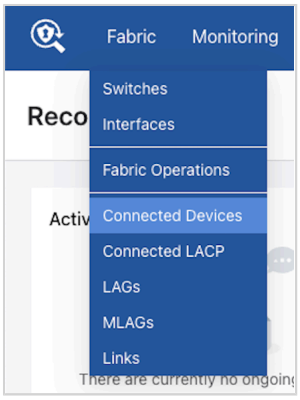
2. To add a new RN, select the **Add Recorder Node** (**Recorder Nodes** tab active) in the **Inventory** section.

Figure 6-14: Provision Recorder Node



3. Enter the following information in the required fields:
   - Assign a name to the RN.
   - Set the **MAC Address** of the RN. Obtain the MAC address from the chassis ID of the connected device, using the **Fabric** > **Connected Devices**.

     Figure 6-15: Connected Devices



4. Configure the following options as needed:
   - **Recording**: Recording is enabled by default. To disable recording on the RN, move the **Recording** toggle switch to **Off**. When recording is enabled, the RN records the matching traffic directed from the filter interface defined in a DMF policy.
   - **Disk Full Policy**: The default packet removal policy is **Continuous** operating as FIFO (First In First Out), which means the oldest packets are deleted to make room for newer packets. This occurs only when the RN disks are full. The alternative removal policy is **Stop**, which causes the RN to stop recording when the disks are full and wait until disk space becomes available. Disk space can be made available by leveraging the RN delete operation to remove all or selected time ranges of recorded packets.
   - **Backup Disk Policy**: Specify the disk backup policy to as desired. Select from one of the three following options:
     - **No Backup**: This is the default option and is also the recommended option when no extra disk is available.
     - **Remote Extend**: In this option, recording is performed on the local disks. When full, the recording continues on a remote Isilon cluster mounted over NFS. In this mode, the remote disks are called **backup disks**. With regard to the *Disk Full Policy*, if set to:
       - **Stop**: Recording stops when both local and remote disks become full.
       - **Continuous**: When the configured threshold is reached, the oldest files from both disks are removed until the disk usage returns below the threshold number.
     - **Local Fallback**: In this option, recording is performed on a remote Isilon cluster mounted over NFS. If the connection between the Recorder Node and the remote cluster fails, the recording is performed on the local disks until the failure is resolved. In this mode, the local disks are called **backup disks**. With regard to the *Disk Full Policy*, if set to:
       - **Stop**: Recording stops when the remote disks become full.
       - **Continuous**: When the configured threshold is reached, the oldest files from both disks are removed until the disk usage returns below the threshold number.

> **Note:** A connection failure should not occur due to a misconfiguration of the NFS server on the DMF Controller. In such cases, the recording stops until the Controller's configuration is fixed.

- **Max Packet Age**: Change the **Max Packet Age** to set the maximum number of minutes that recorded packets are kept on the RN. Packets recorded are discarded after the specified number of minutes. This defines the maximum age in minutes of any packet in the RN. It can be used in combination with the **Disk Full Policy** to control when packets are deleted based on age rather than disk utilization alone. When unset, **Max Packet Age** is not enforced.
- **Incident Look Ahead Buffer**: Assign the number of minutes the RN pre-buffer allows for windowed retention of packets received by the RN for a specified length of time. By default, the **Incident Look Ahead Buffer** is set to zero minutes (disabled). With a nonzero **Incident Look Ahead Buffer** setting and triggering a recorder event, any packets in the pre-buffer are saved to disk, and any packets received by the recorder after the trigger are saved directly to disk. When terminating an ongoing recorder event, a new pre-buffer is established in preparation for the next event.
- **Max Disk Utilization**: Specify the maximum utilization allowed on the index and packet disks. The **Disk Full Policy** is enforced at this limit. If left unset, then the disks space are used to capacity.
- **Parse MetaWatch Trailer**: Determine the parsing of the MetaWatch trailer.
  - **Off**: When set to **Off**, the RN will not parse the MetaWatch trailer, even if it is present in incoming packets.
  - **Auto**: When set to **Auto**, the RN will look for a valid timestamp in the last 12 bytes of the packet. If it matches the system timestamp closely enough, the trailer is parsed by the RN.
  - **Always**: When set to **Always**, RN will assume the last 12 bytes of packet is a MetaWatch trailer and parse it, even if it did not find a valid timestamp.
- **Indexing**: All the indexing options are enabled by default. To disable any of the indexing behaviors, select **Indexing** and deselect from the list, as required. These include:
  - MAC Source
  - MAC Destination
  - VLAN 1
  - VLAN 2
  - VLAN 3
  - IPv4 Source
  - IPv4 Destination
  - IPv6 Source
  - IPv6 Destination
  - IP Protocol
  - Port Source
  - Port Destination

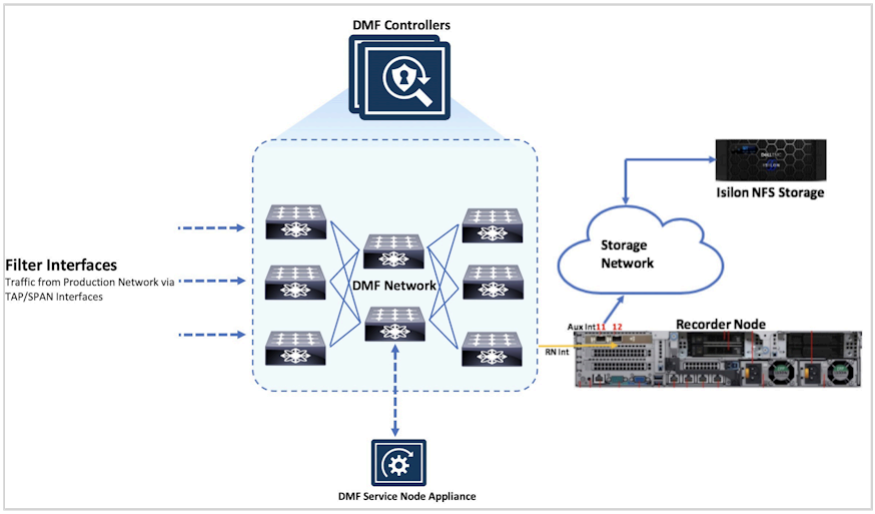
- MPLS
- Community ID
- MetaWatch Device ID
- MetaWatch Port ID

For more details, see the Indexing Configuration section.

5. Configure network and storage settings selecting **Network & Storage Config**.

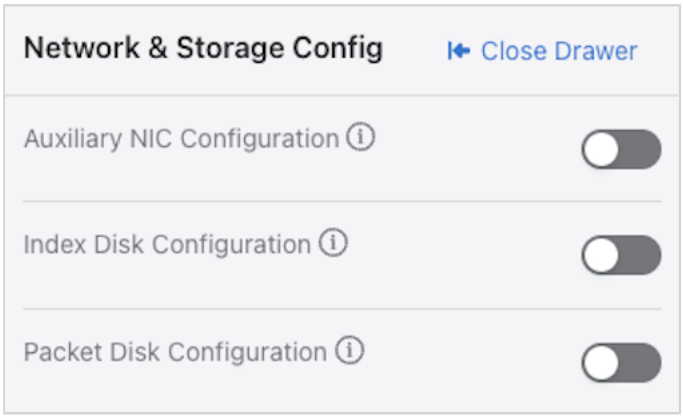
   In order to store packets on external storage using an NFS mount, connect the Recorder Node's (RN) auxiliary interface to the same network and subnet where the NFS storage resides, as displayed in the figure below.

   Figure 6-16: Topology to Use External Storage



> **Note:** Create the volume for the index and packet on the NFS storage first. Refer to the vendor-specific NFS storage documentation about creating the volume (or path).

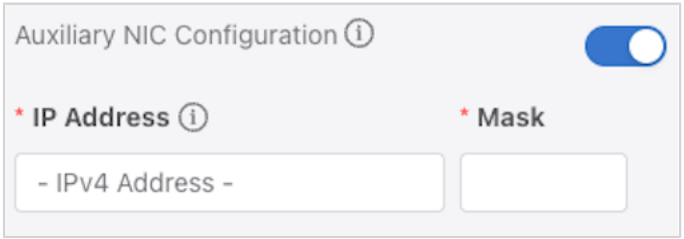
Figure 6-17: Network & Storage Settings



a. **Auxiliary NIC Configuration**: Move the toggle switch to **On**.

   Enter the **IP Address** and **Mask**. These are required fields.

   Figure 6-18: Auxiliary NIC Configuration



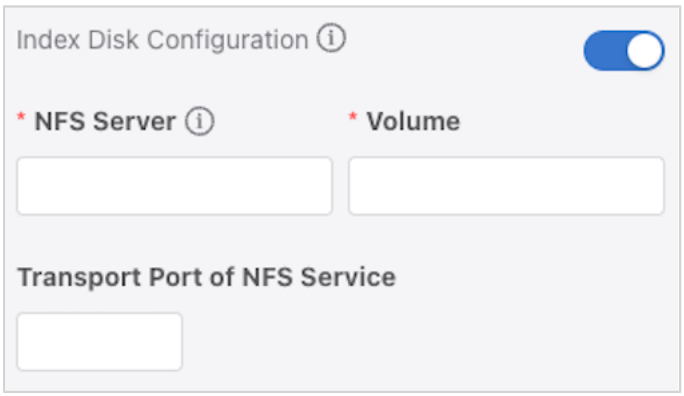
b. **Index Disk Configuration**: Move the toggle switch to **On**.

Enter the **NFS Server** as an IP address or hostname. For external NFS storage, such as Isilon, connect the auxiliary interface of the RN to a network and subnet that is reachable to Isilon NFS storage. When recording to an Isilon cluster over NFS with SmartConnect, the name of the storage pool can be specified. SmartConnect requires this name be registered as an A record pointing to the SmartConnect VIP in the DNS server used by the recorder node. The recorder will establish many mount points to the provided volume name in this storage pool. Each mount point will resolve to a different node in the storage pool, allowing the recorder to distribute recorded packets and metadata across multiple nodes in parallel.

Enter the **Volume**: Specify the storage location or path on the NFS server for the **Index Disk**.

Enter the **Transport Port of NFS Service** details. DMF uses the default value (2049) if no value is specified. Specify a value if the NFS storage has been configured to use something other than the default value.

Figure 6-19: Index Disk Configuration



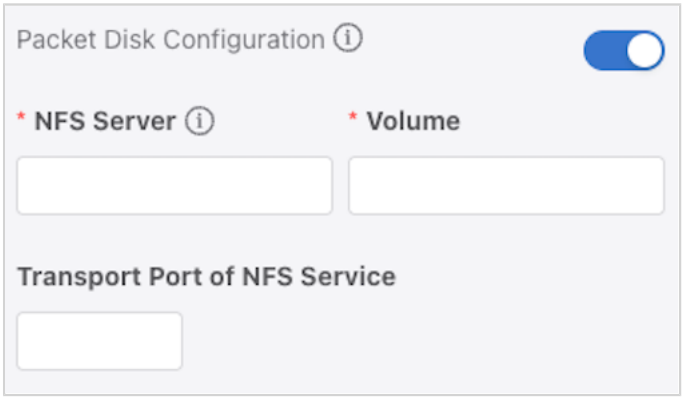
c. **Packet Disk Configuration**: Move the toggle switch to **On**.

Enter the **NFS Server** as an IP address or hostname. For external NFS storage, such as Isilon, connect the auxiliary interface of the RN to a network and subnet that is reachable to Isilon NFS storage. When recording to an Isilon cluster over NFS with SmartConnect, the name of the storage pool can be specified. SmartConnect requires this name be registered as an A record pointing to the SmartConnect VIP in the DNS server used by the recorder node. The recorder will establish many mount points to the provided volume name in this storage pool. Each mount point will resolve to a different node in the storage pool, allowing the recorder to distribute recorded packets and metadata across multiple nodes in parallel.

Enter the **Volume**: Specify the storage location or path on the NFS server for the **Packet Disk**.

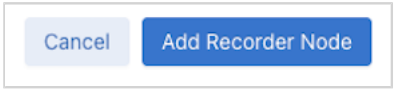
Enter the **Transport Port of NFS Service** details. DMF uses the default value (2049) if no value is specified. Specify a value if the NFS storage has been configured to use something other than the default value.

Figure 6-20: Packet Disk Configuration



6. Select **Add Recorder Node** to save and close the configuration page.

Figure 6-21: Provision Recorder Node-Indexing



Note: When editing the configuration of a previously added RN to use external storage versus local storage or vice versa, **reboot** the RN.

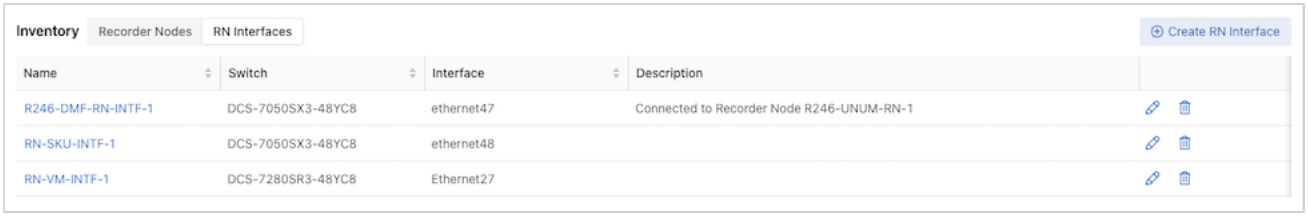
```
dmf-controller-1># restart recorder-node rn-name application
```

6.6.3 Configure a Recorder Node Interface

To record packets to a recorder node using a DANZ Monitoring Fabric (DMF) policy, configure a DMF Recorder Node (RN) interface that defines the switch and interface in the monitoring fabric where the RN is connected. The DMF RN interface is referenced by name in the DMF policy as the destination for traffic matched by the policy. To configure a DMF RN interface, perform the following steps:

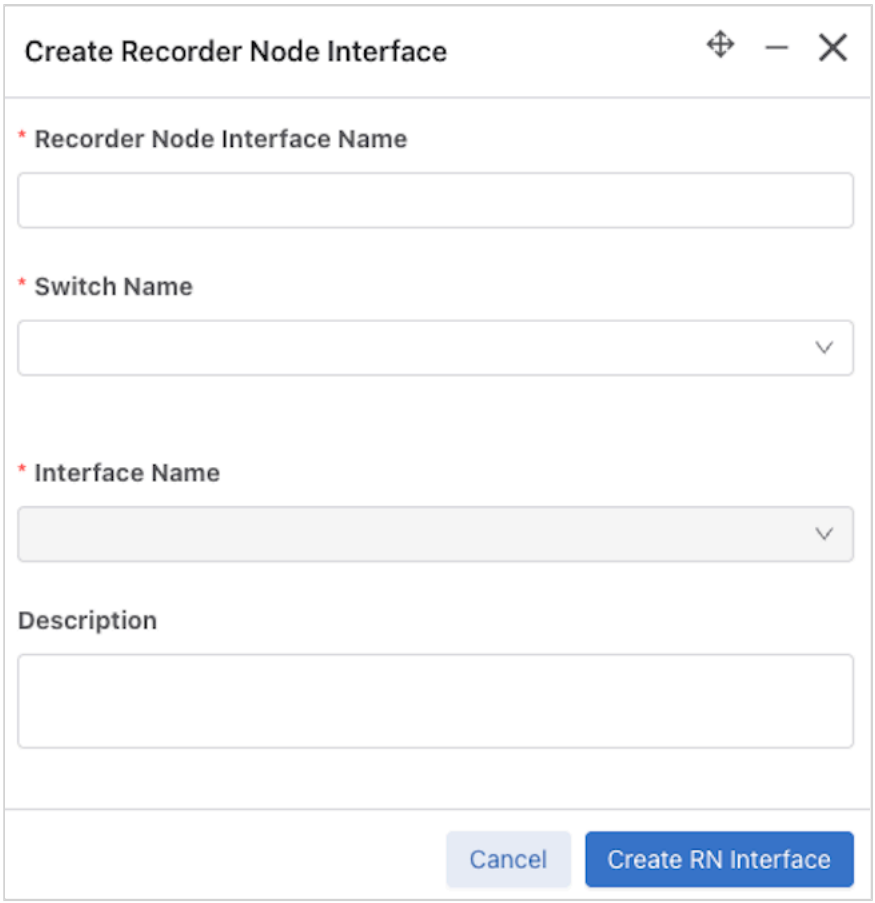
1. To add a new RN interface, select the **RN Interfaces** tab in the **Inventory** section.

   Figure 6-22: Create DMF Recorder Node Interface



2. Select **Create RN Interface**.

   Figure 6-23: Create Recorder Node Interface



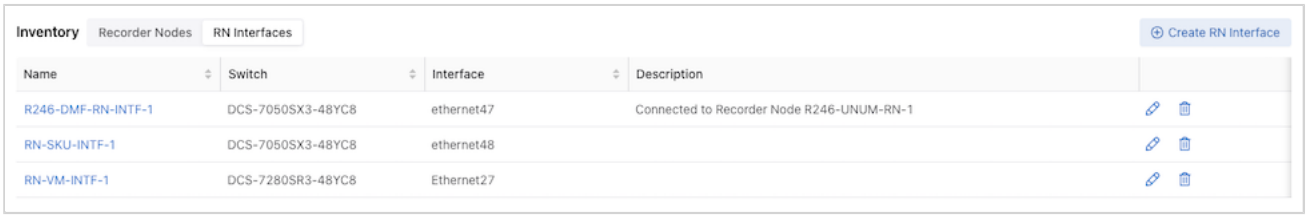
3. Enter or select the following parameters:
   - **Recorder Node Interface Name**: Assign a name to the interface.
   - **Switch Name**: Select a switch from the drop-down list containing the interface connecting the RN to the monitoring fabric.
   - **Interface Name**: Select an interface from the drop-down list connecting the RN to the monitoring fabric.
   - **Description**: (optional) Enter a description of the RN interface.
4. Select **Create RN Interface** to add the configuration to the DMF Controller.

6.6.4 Edit a Recorder Node Interface

To edit a DMF RN interface, perform the following steps:

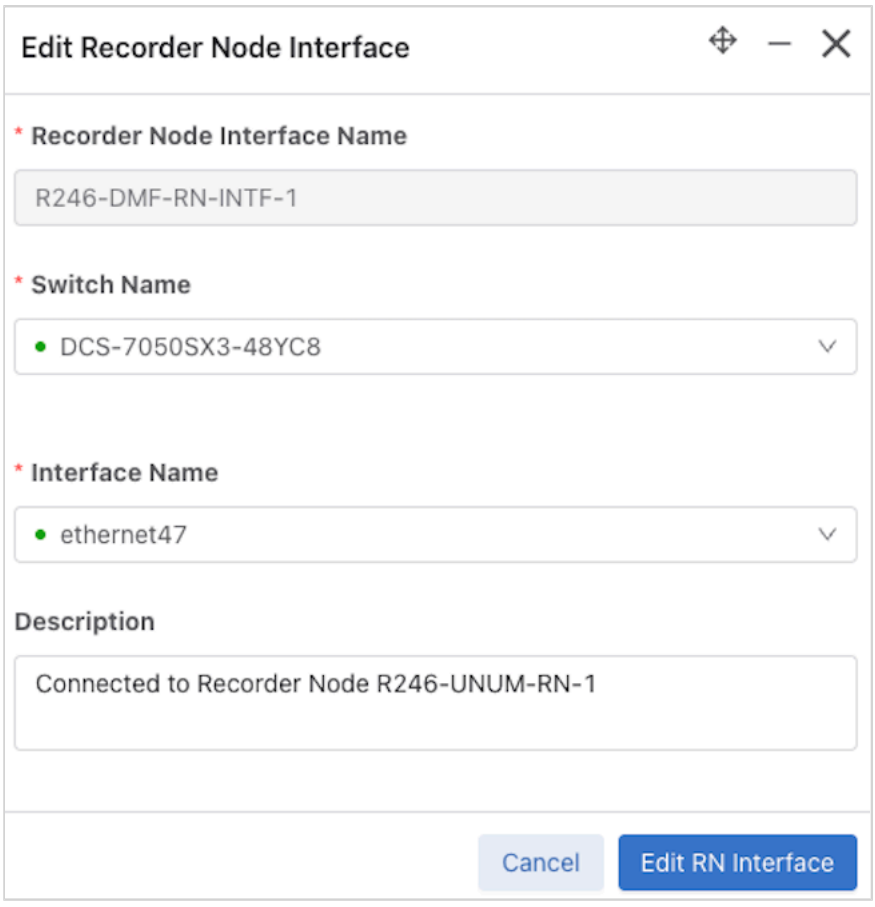
1. Select the **RN Interfaces** tab in the **Inventory** section.

   Figure 6-24: Create DMF Recorder Node Interface



2. Select an interface from the **Inventory** list and use the **Edit** icon.

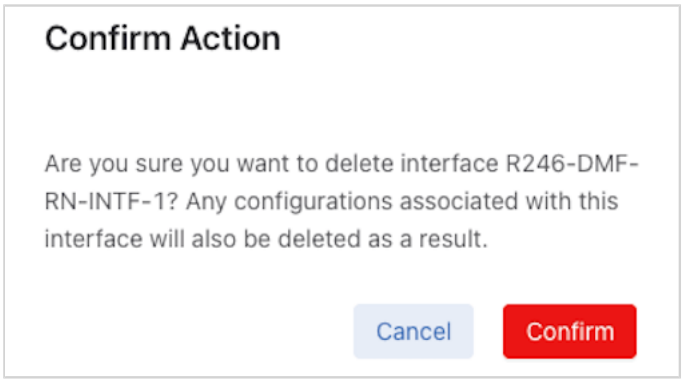
   Figure 6-25: Create Recorder Node Interface



3. Edit the following parameters:
   - **Switch Name**: Select a switch from the drop-down list containing the interface connecting the RN to the monitoring fabric.
   - **Interface Name**: Select an interface from the drop-down list connecting the RN to the monitoring fabric.
   - **Description**: (optional) Edit the description of the RN interface.

4. Select **Edit RN Interface** to modify and update the interface configuration.
5. To delete an interface, use the **Delete** icon and **Confirm**.

Figure 6-26: Confirm Delete

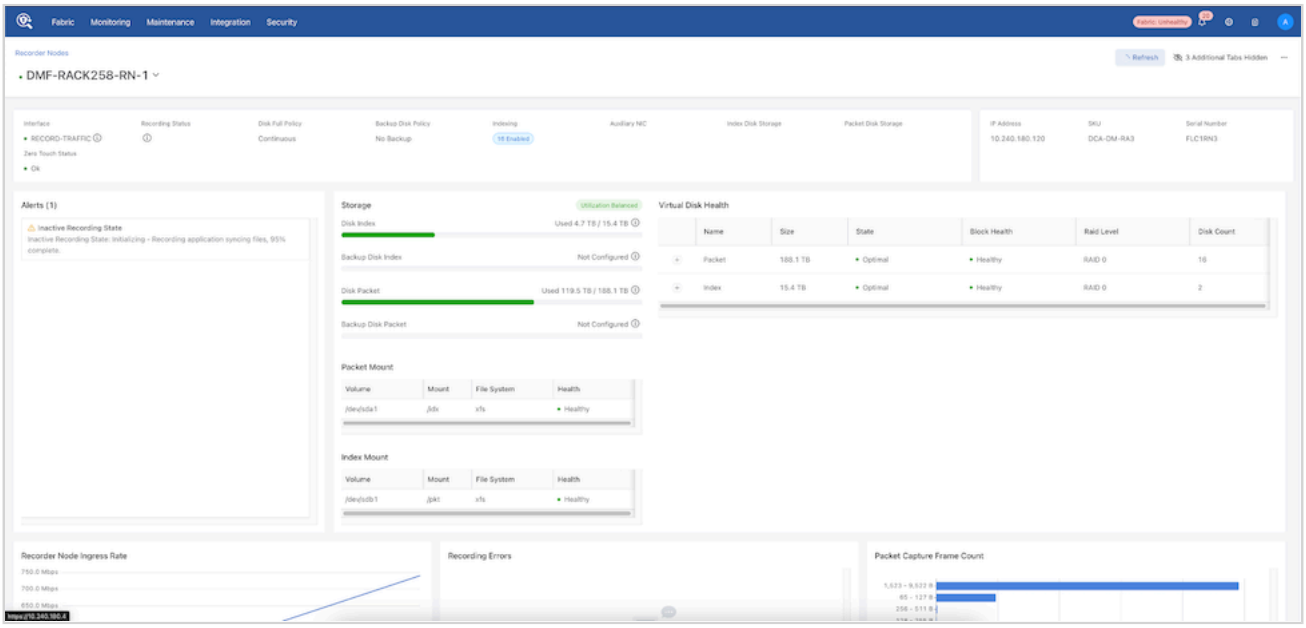


6.6.5 Recorder Node Details

The **Recorder Node Detail** page displays specific information about the **Recorder Node**.

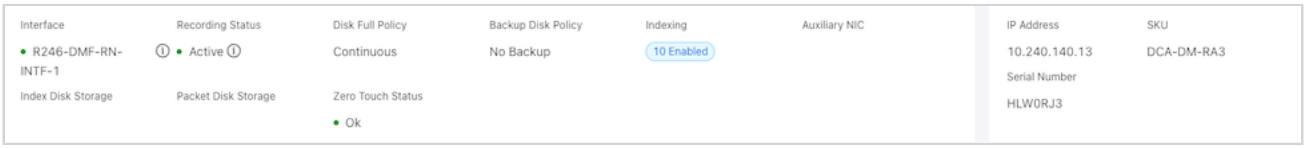
In the **Inventory** window (**Recorder Nodes** tab active), select the Recorder Node **Name** to navigate to the details page.

Figure 6-27: Recorder Nodes Details



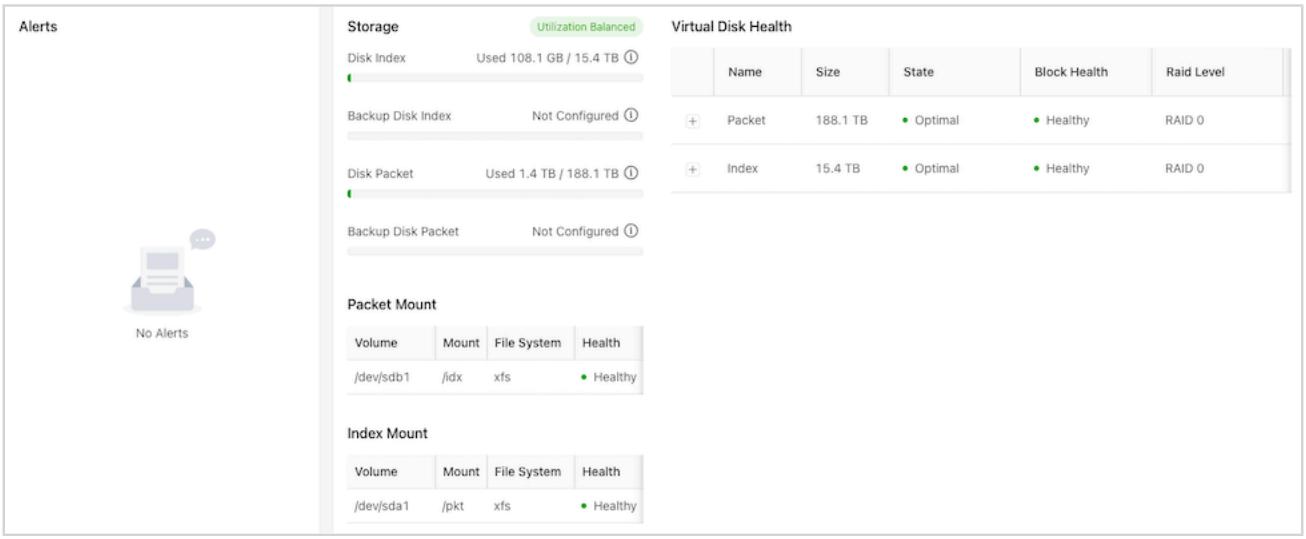
The upper dashboard displays a brief summary of the status of the Recorder Node. A **Recorder Nodes** link returns to the **Recorder Nodes** dashboard and a drop-down speed navigation to other RN instances.

Figure 6-28: Brief Summary



The middle dashboard displays any active **Alerts** on the selected **Recorder Node** and **Storage** details:

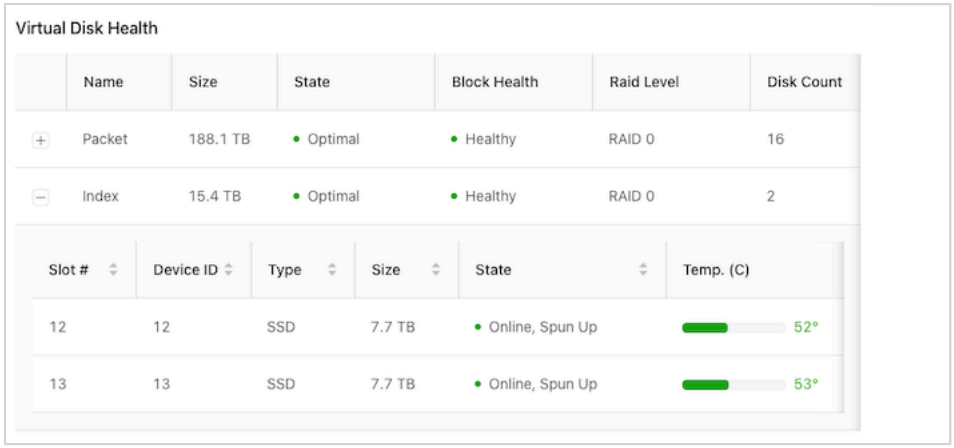
Figure 6-29: Alerts and Storage



- **Storage**
  - Disk Index
  - Backup Disk Index
  - Disk Packet
  - Backup Disk Packet
- **Index Mount**
  - Volume, Mount, File System, and Health
- **Packet Mount**
  - Volume, Mount, File System, and Health
- **Virtual Disk Health**
  - Name, Size, State, Block Health, and Raid Level
  - Use the **+** icon to obtain granular information on the Virtual Disk **Packet** and **Index** health.

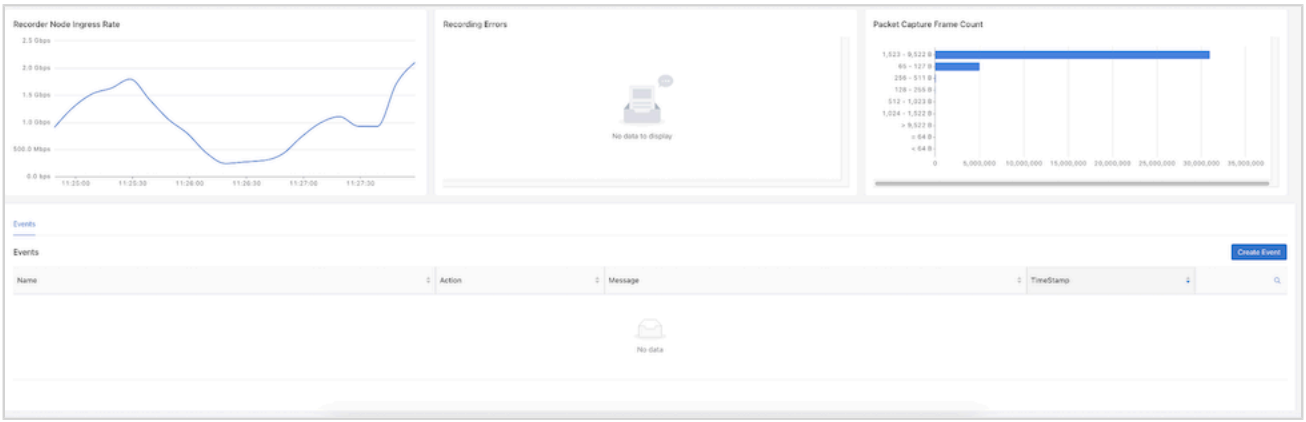
- Slot #, Device ID, Type, Size, State, Temp, Predicted Failure Count, Media Error Count, and Other Error Count.

Figure 6-30: Virtual Disk Health Details



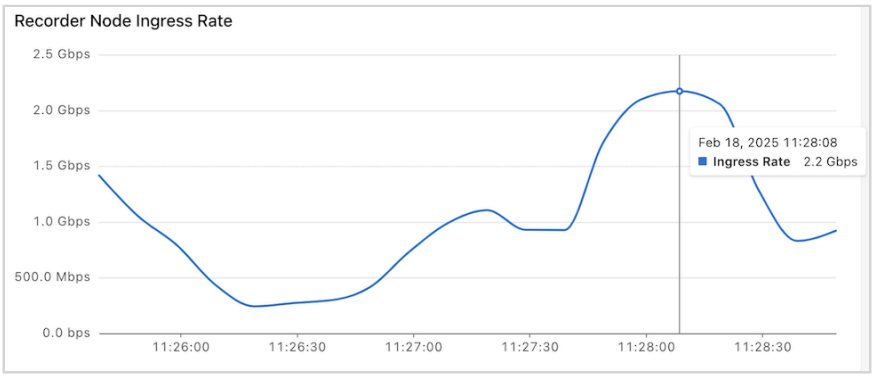
The lower dashboard displays live graphs of the **Recorder Node Ingress Rate**, **Recording Errors**, and **Packet Capture Frame Count**, and **Events**.

Figure 6-31: Recorder Node Rates, Errors and Counts.



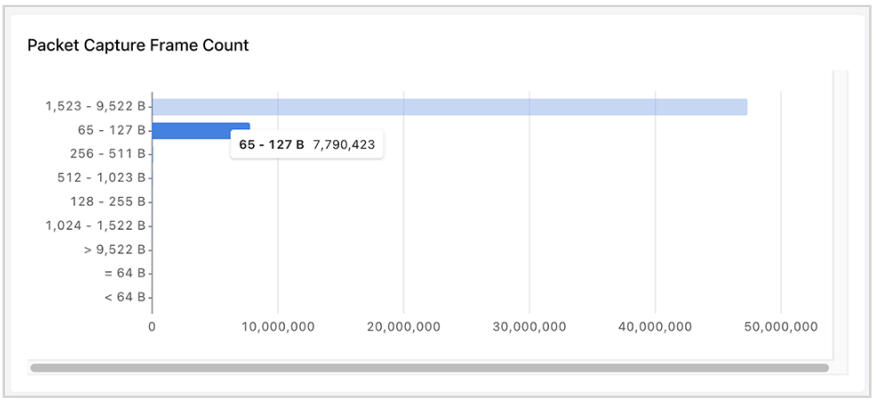
Hovering over the graph displays **Timestamps** and **Ingress Rate**.

Figure 6-32: Timestamps and Ingress Rate



Hovering over a bar displays **Timestamps** and **Frame Count**.

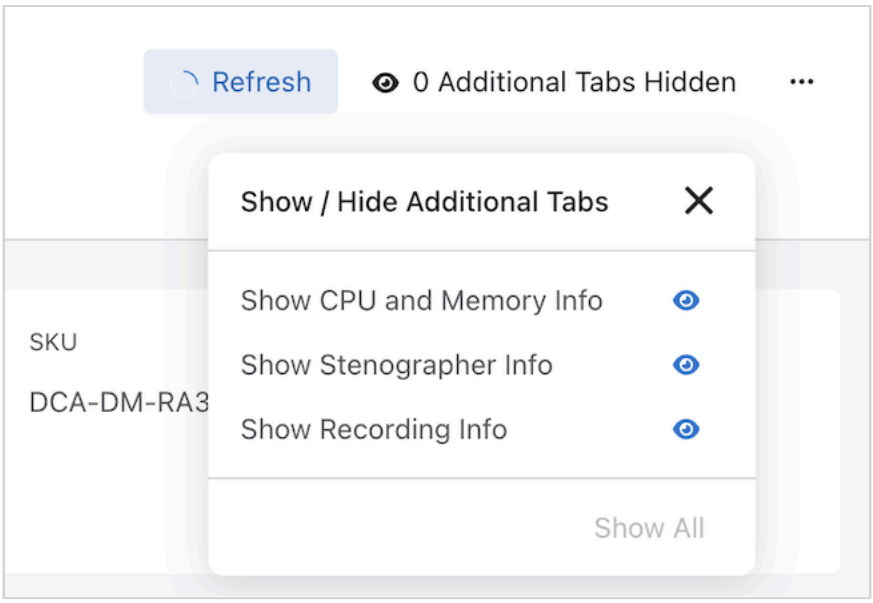
Figure 6-33: Timestamps and Frame Count



Show / Hide Additional Tabs determine the information displayed in the **Event** section. They are selectable and persist until changed. The UI displays up to four tabs.

Note: The **Events** table cannot be hidden.

Figure 6-34: Show / Hide Additional Tabs



6.6.6 Recorder Node Events

The Recorder Node Details dashboard displays **Events** in the lower section.

Figure 6-35: Recorder Nodes Details

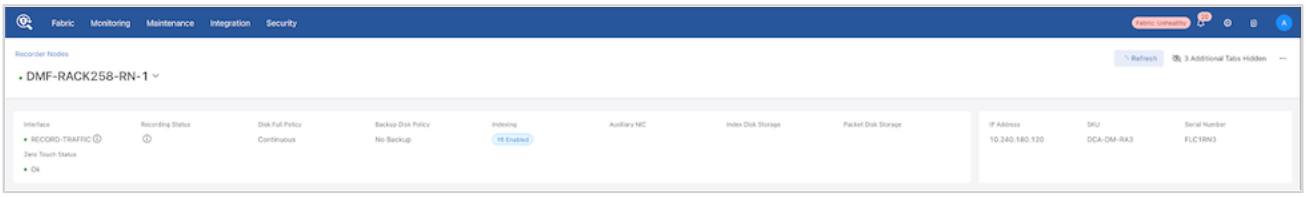
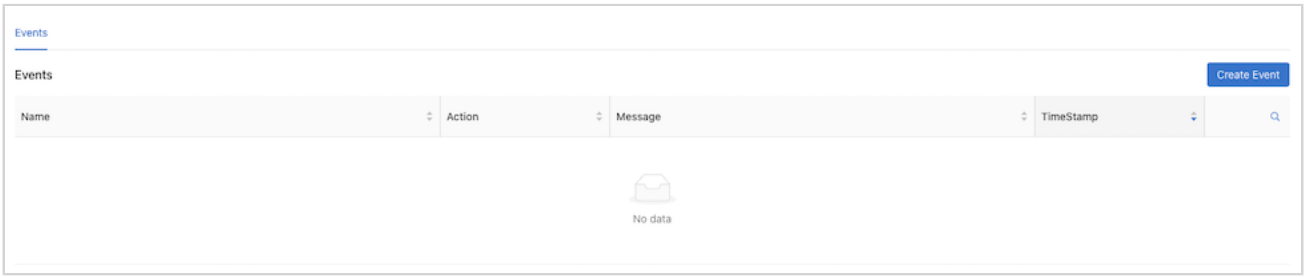


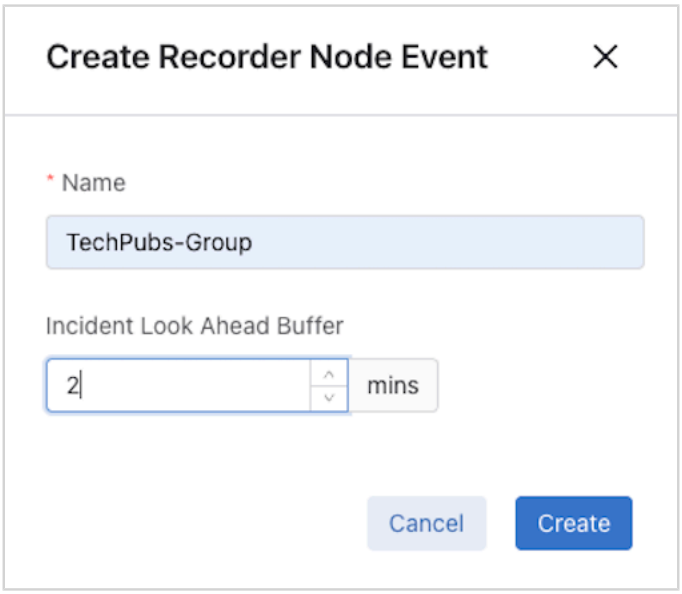
Figure 6-36: Events Dashboard



Select **Create Event** to create an event. Enter the required information:

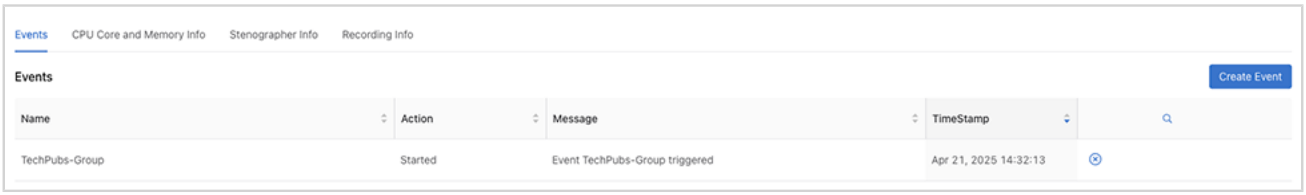
- **Name**: Enter a name for the RN event.
- **Incident Look Ahead Buffer**: Enter a value (in minutes).

Figure 6-37: Create Recorder Node Event



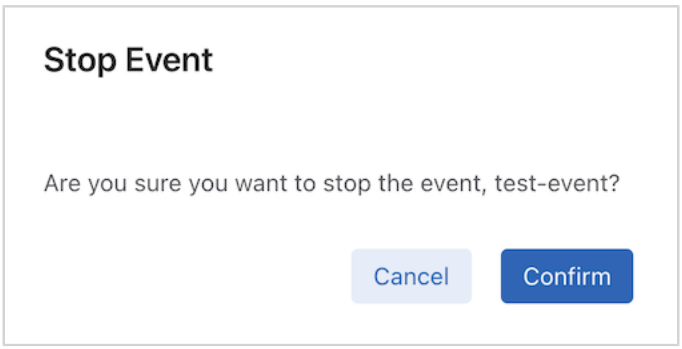
Select **Create** to create a **Recorder Node Event**.

Figure 6-38: Added Event



Use the **X** icon to stop the event and confirm.

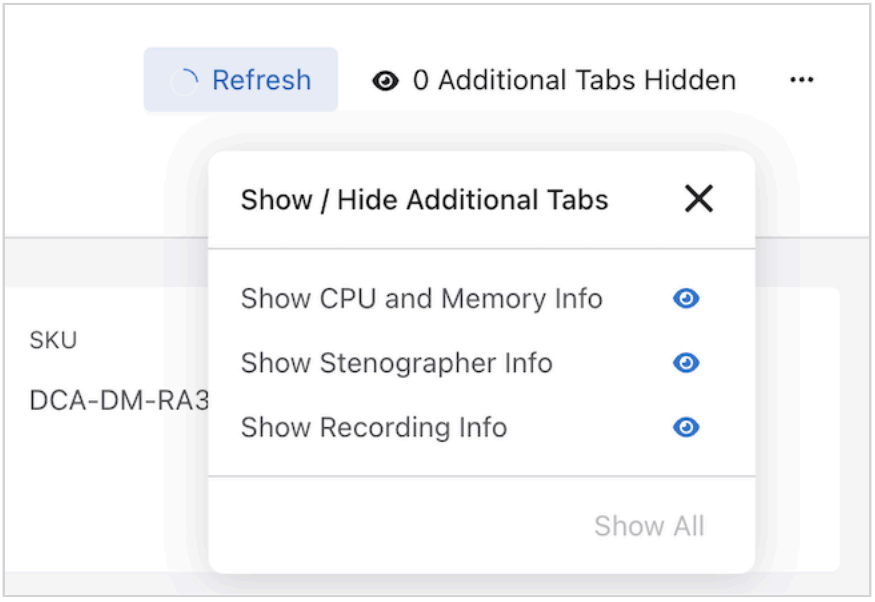
Figure 6-39: Confirm Stop



Show / Hide Additional Tabs determine the information displayed in the **Event** section. They are selectable and persist until changed. The UI displays up to four tabs.

Note: The **Events** table cannot be hidden.

Figure 6-40: Show / Hide Additional Tabs



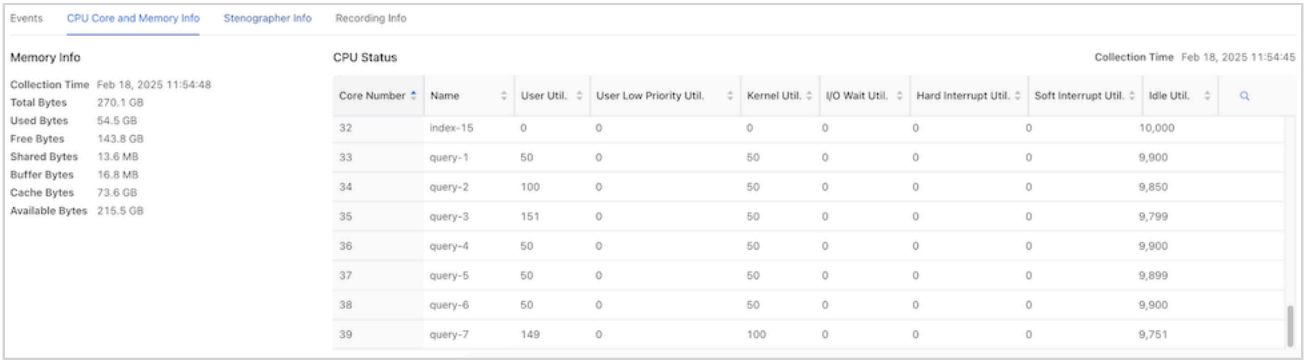
Additional tabs include:

- **CPU Core and Memory Info**
- **Stenographer Info**
- **Recording Info**

6.6.6.1 CPU Core and Memory Info

The **CPU Core and Memory Info** dashboard displays **Memory Info** and **CPU Status**.

Figure 6-41: CPU Core and Memory Info



Memory Info summary information includes:

- Collection Time
- Total Bytes
- Used Bytes
- Free Bytes
- Shared Bytes
- Buffer Bytes
- Cache Bytes
- Available Bytes

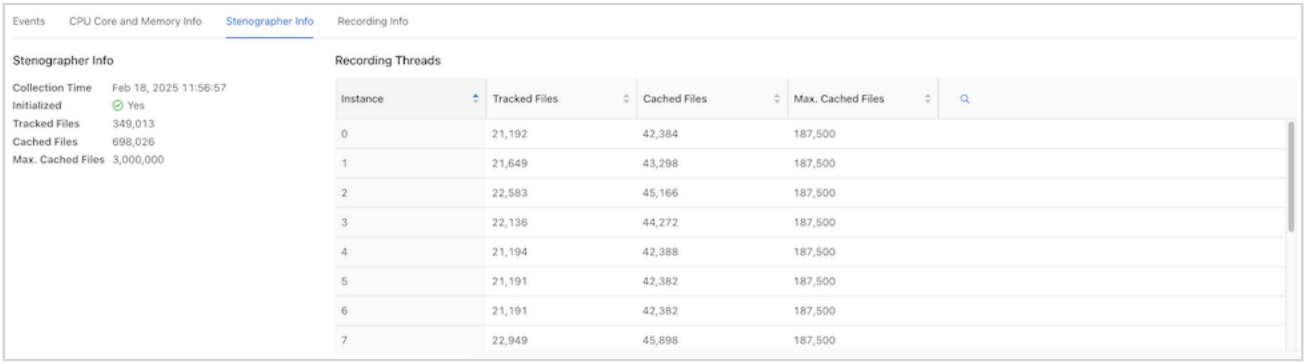
CPU Status tabular information includes:

- Core Number
- Name
- User Utilization
- User Low Priority Utilization
- Kernel Utilization
- I/O Wait Utilization
- Hard Interrupt Utilization
- Soft Interrupt Utilization
- Idle Utilization

6.6.6.2 Stenographer Info

The **Stenographer Info** dashboard displays **Stenographer Info** and **Recording Threads**.

Figure 6-42: CPU Core and Memory Info



Stenographer Info summary information includes:

- Collection Time
- Initialized
- Tracked Files
- Cached Files
- Maximum Cached Files

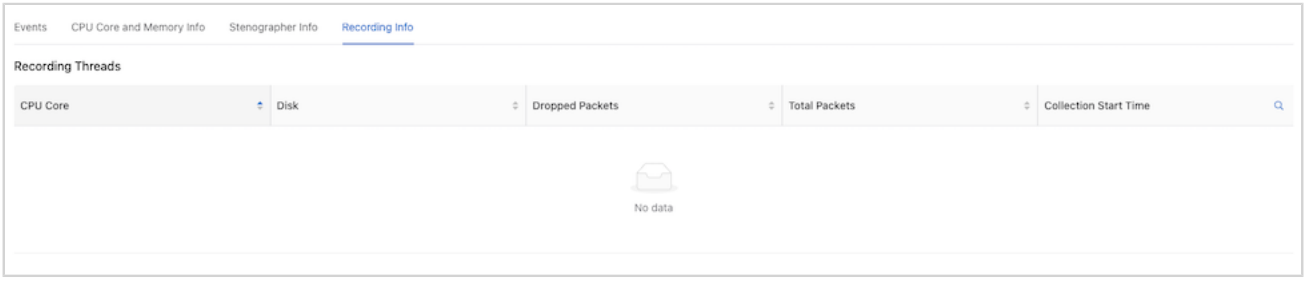
Recording Threads tabular information includes:

- Instance
- Tracked Files
- Cached Files
- Maximum Cached Files

6.6.6.3 Recording Info

The **Recording Info** dashboard displays **Recording Threads**.

Figure 6-43: Recording Info



Recording Threads tabular information includes:

- CPU Core
- Disk
- Dropped Packets
- Total Packets
- Collection Start Time

6.6.7 Assign a Recorder Node Interface to a Policy

To forward traffic to a Recorder Node (RN), include one or more RN interfaces as a delivery interface in a DANZ Monitoring Fabric (DMF) policy. Two methods exist to create a Policy:

Using Recorder Nodes to Create a Policy

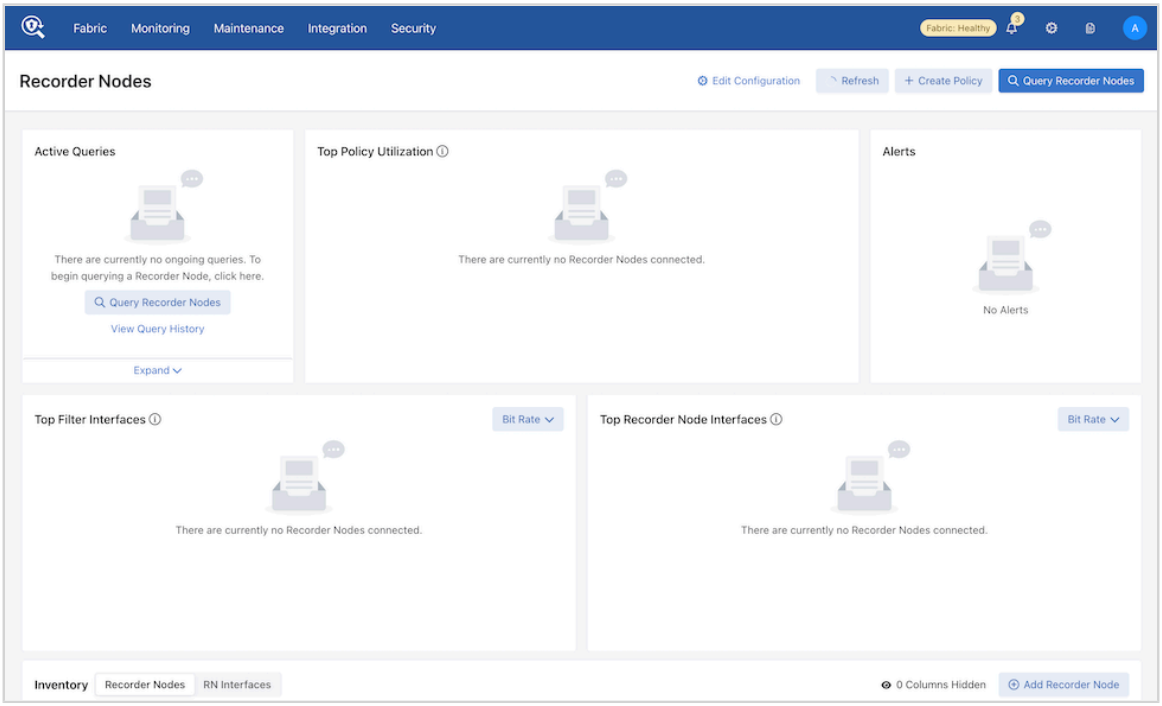
Using Monitoring Policies to Create a Policy

6.6.7.1 Using Recorder Nodes to Create a Policy

When creating a new policy or editing an existing policy, select the RN interfaces from the **Monitoring** > **Recorder Nodes**.

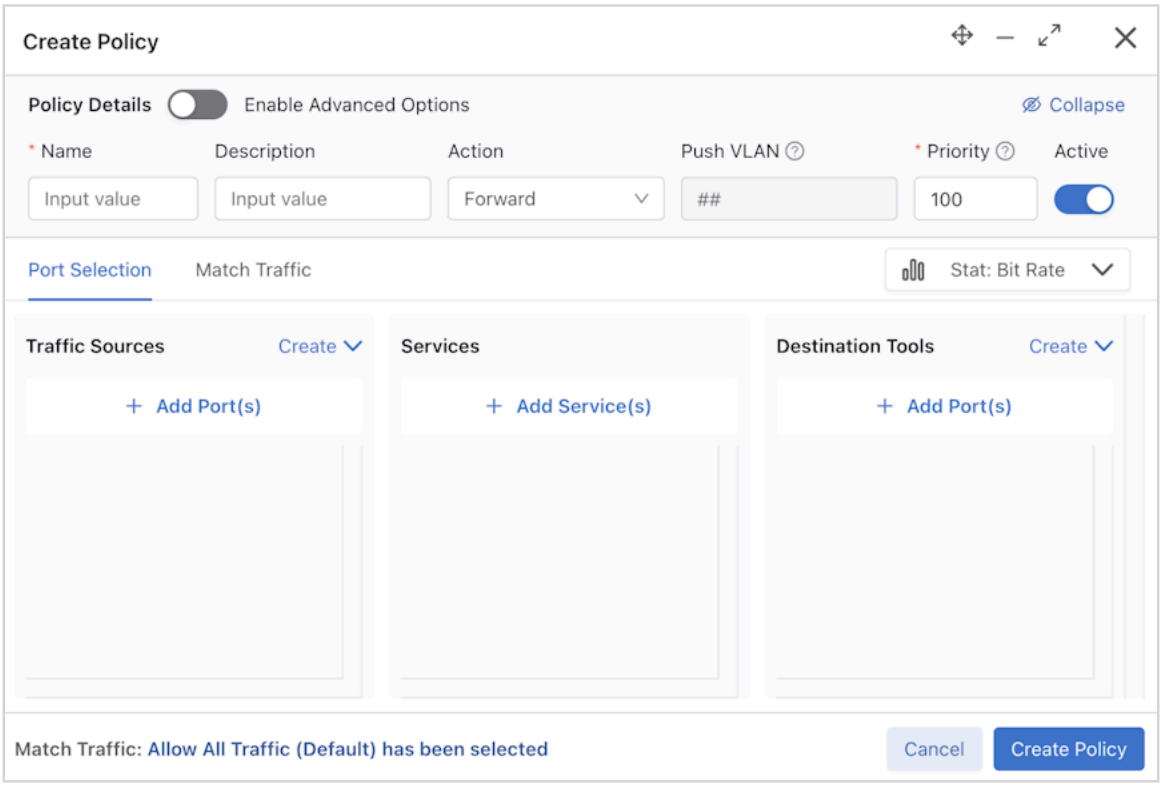
> **Note:** For more information on configuring Policies refer to the Managing DMF Policies section.

Figure 6-44: Recorder Nodes



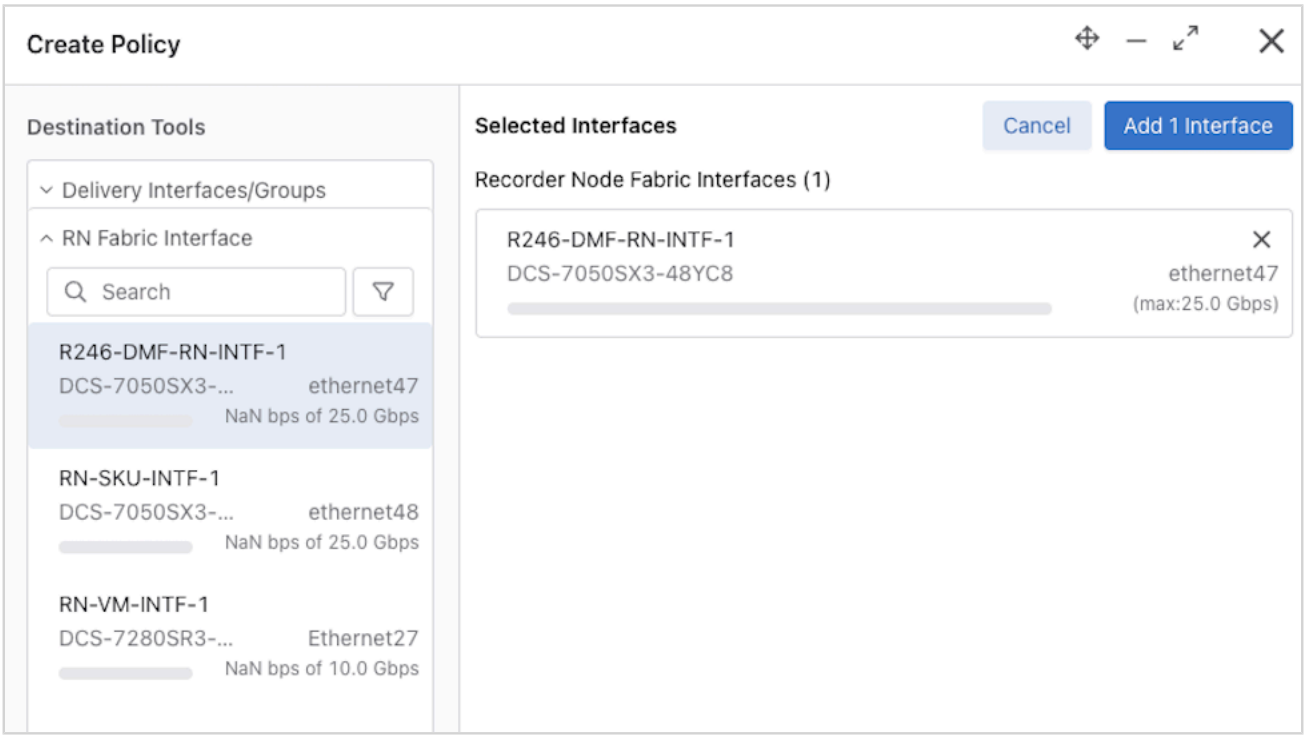
To create a policy, select **+ Create Policy** followed by **Destination Tools** > **Add Ports(s)**.

Figure 6-45: Recorder Node - Create Policy



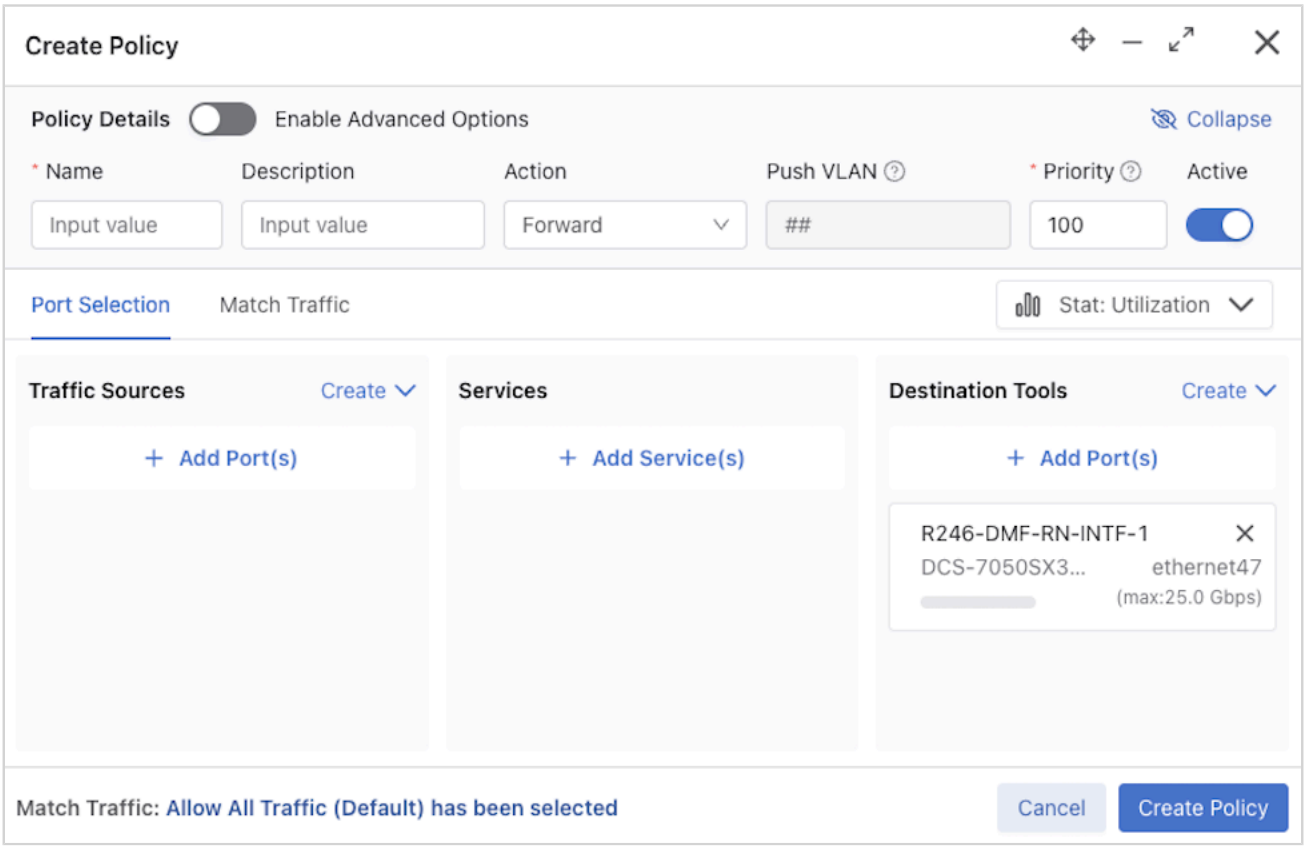
Use **RN Fabric Interface** to select a previously configured RN interface. Select or drag the **Interfaces** or **Recorder Nodes**.

Figure 6-46: Selected Interface



Select **Add n Interface** to add to **Destination Tools**.

Figure 6-47: Destination Tools



Select **Create Policy**.

Note: The RN interface can only be selected and not created in the create policy dialogue.

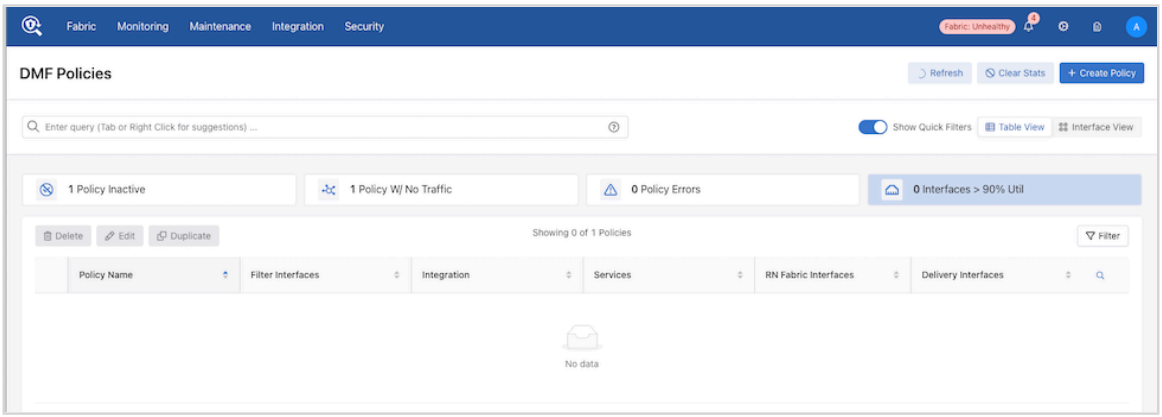
For more information on configuring Policies refer to the Managing DMF Policies section.

6.6.7.2 Using Monitoring Policies to Create a Policy

When creating a new policy or editing an existing policy, select the RN interfaces from the **Monitoring** > **Policies** dialog, as shown in the following screen.

Note: For more information on configuring Policies refer to the Managing DMF Policies section.

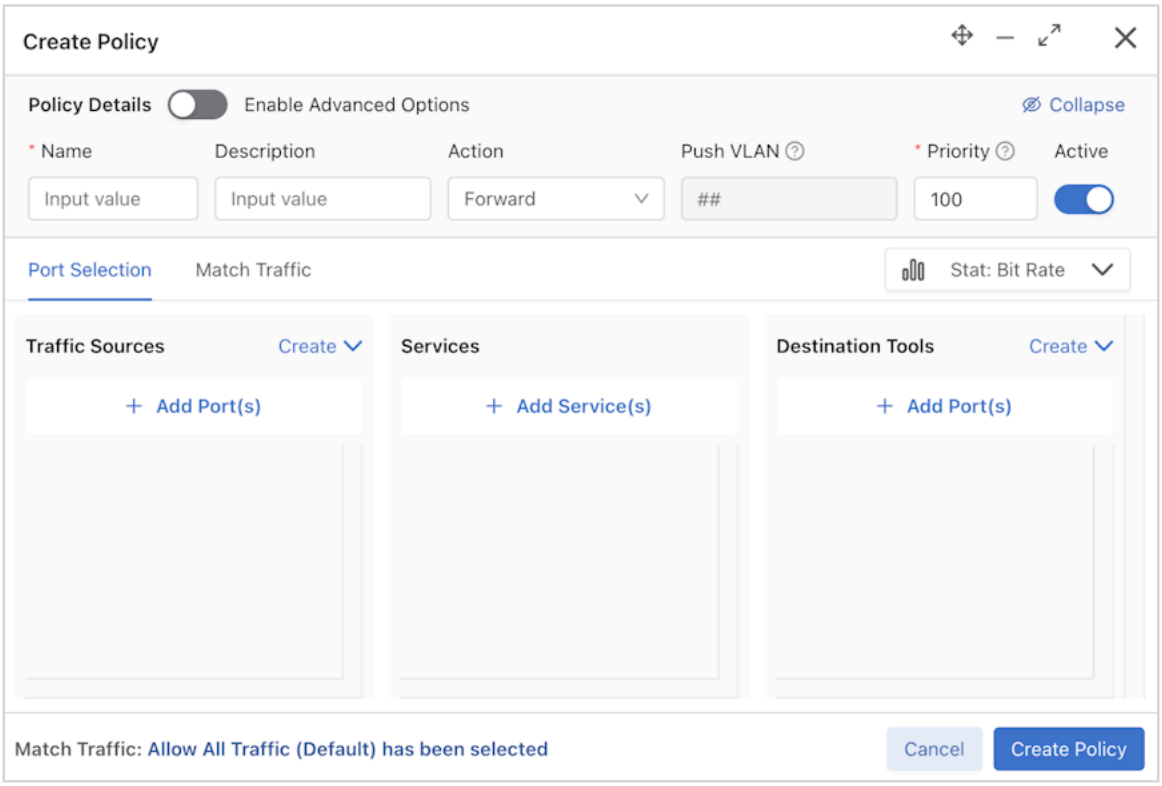
Figure 6-48: DMF Policies



Note: If no RN fabric Interfaces appear, proceed to the **Monitoring** > **Recorder Nodes** > **RN Interfaces** tab in the **Inventory** section to create a RN interface.

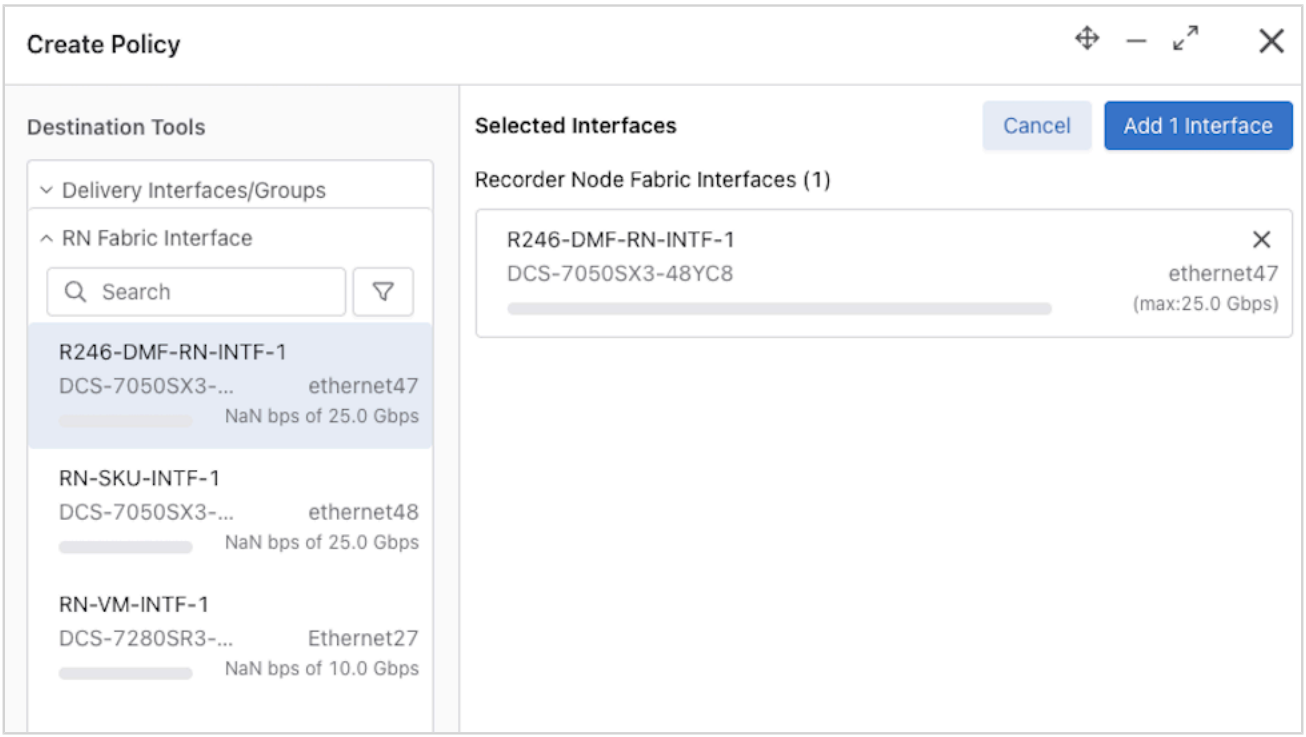
To create a policy, select **+ Create Policy** followed by **Destination Tools** > **Add Ports(s)**.

Figure 6-49: Recorder Node - Create Policy



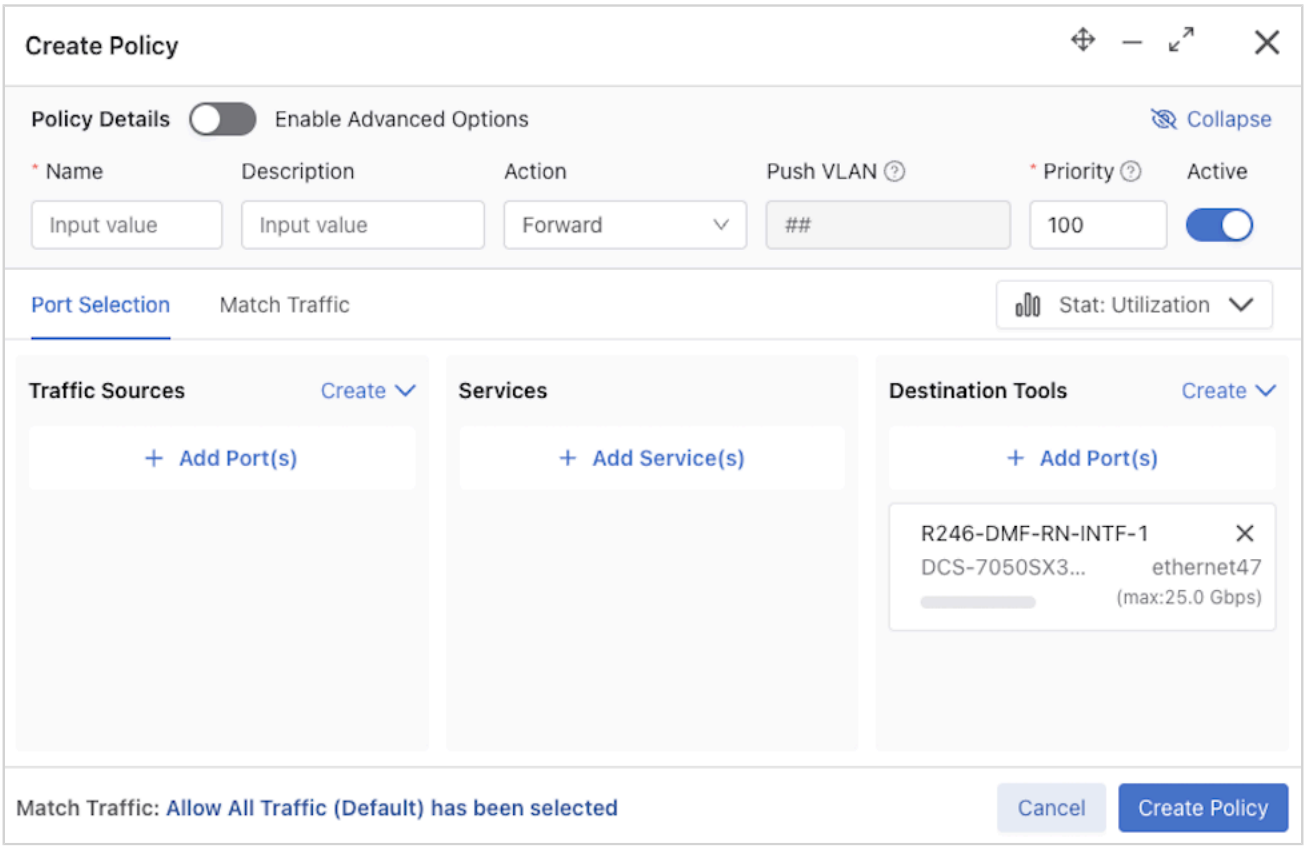
Use **RN Fabric Interface** to select a previously configured RN interface. Select or drag the **Interfaces** or **Recorder Nodes**.

Figure 6-50: Selected Interface



Select **Add n Interface** to add to **Destination Tools**.

Figure 6-51: Destination Tools



Select **Create Policy**.

Note: The RN interface can only be selected and not created in the create policy dialogue.

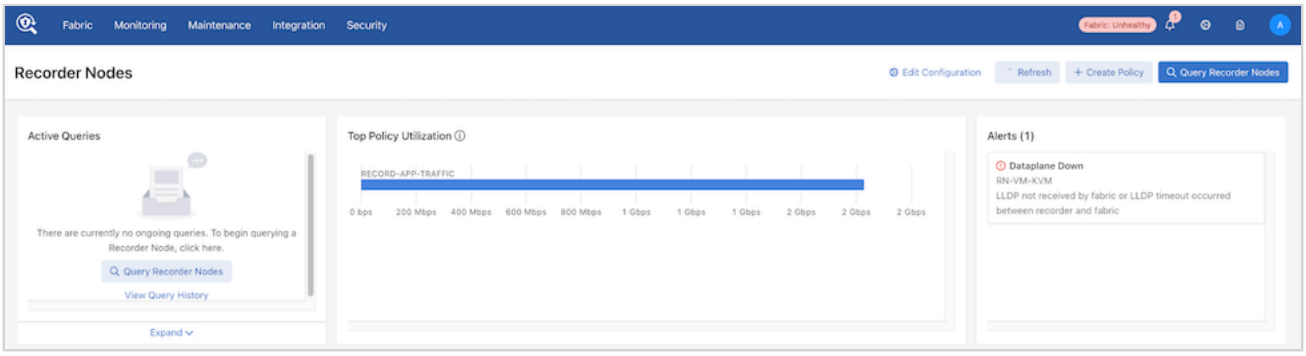
For more information on configuring Policies refer to the Managing DMF Policies section.

6.6.8 Recorder Node Query

Use the options in the **Query Recorder Nodes** section to create a query and submit it to the RN for processing.

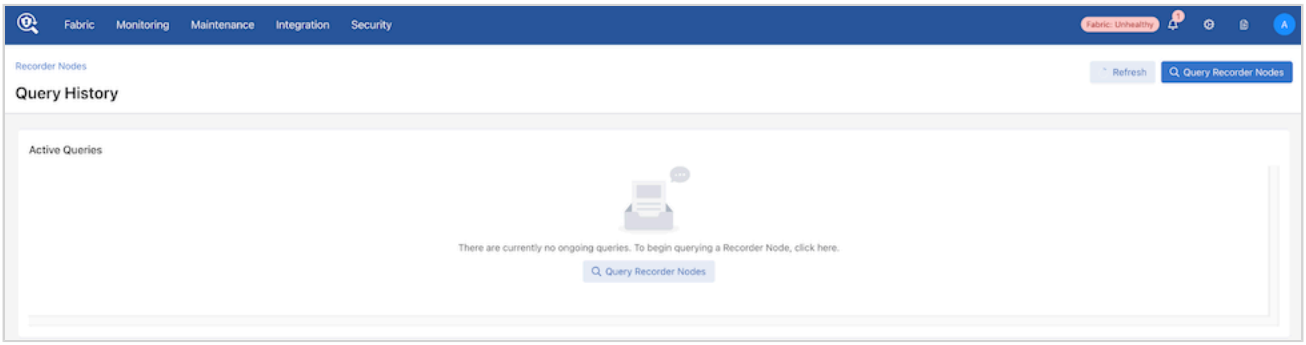
Initiate the **Query Recorder Node** workflow from the **Recorder Nodes** or **Query History** pages.

Figure 6-52: Recorder Nodes



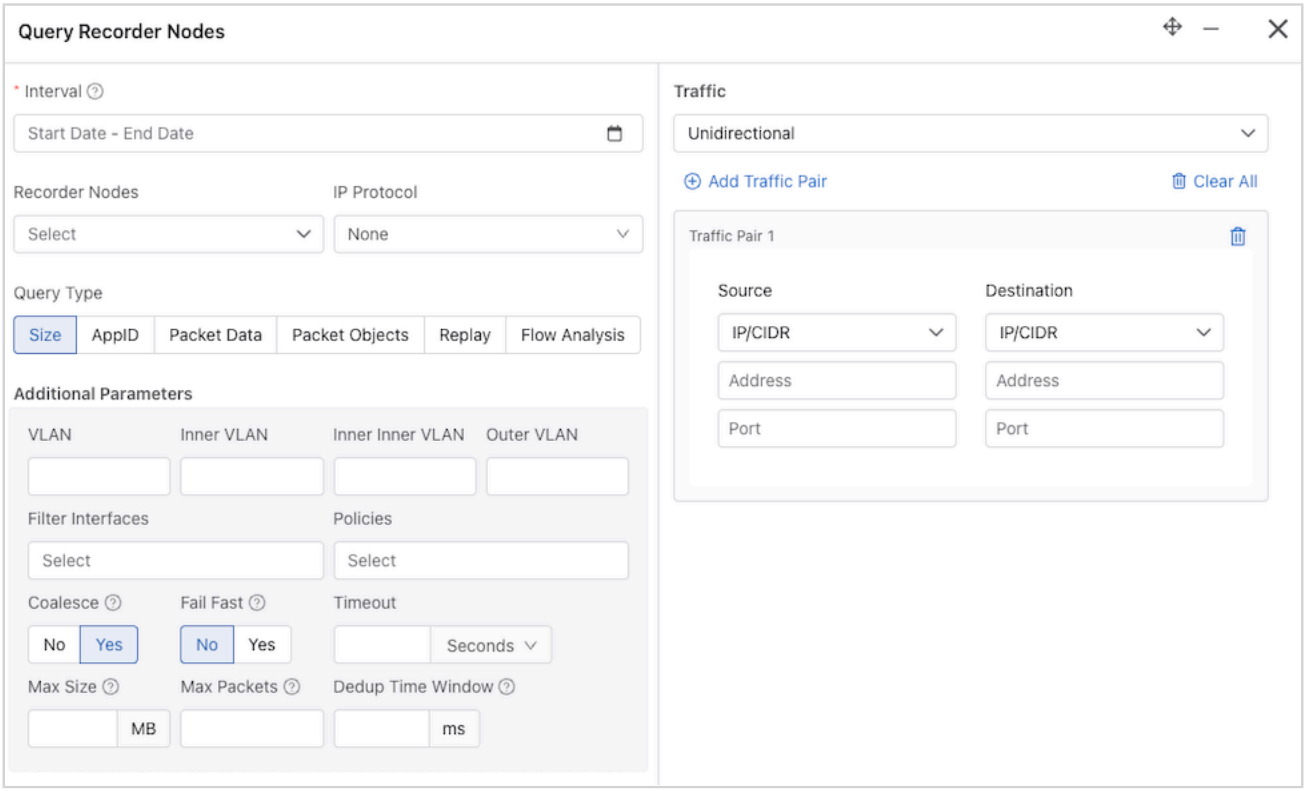
Select **Query Recorder Nodes** to open the **Query Recorder Node** window.

Figure 6-53: Active Queries - Query Recorder Nodes



The Recorder Node (RN) records all the packets received on a filter interface that match the criteria defined in a DMF policy. The RN can recall or analyze recorded packets using various queries. Use the options described in the Query Recorder Node section to create a query and submit it to the RN for processing.

Figure 6-54: Query Action Parameters



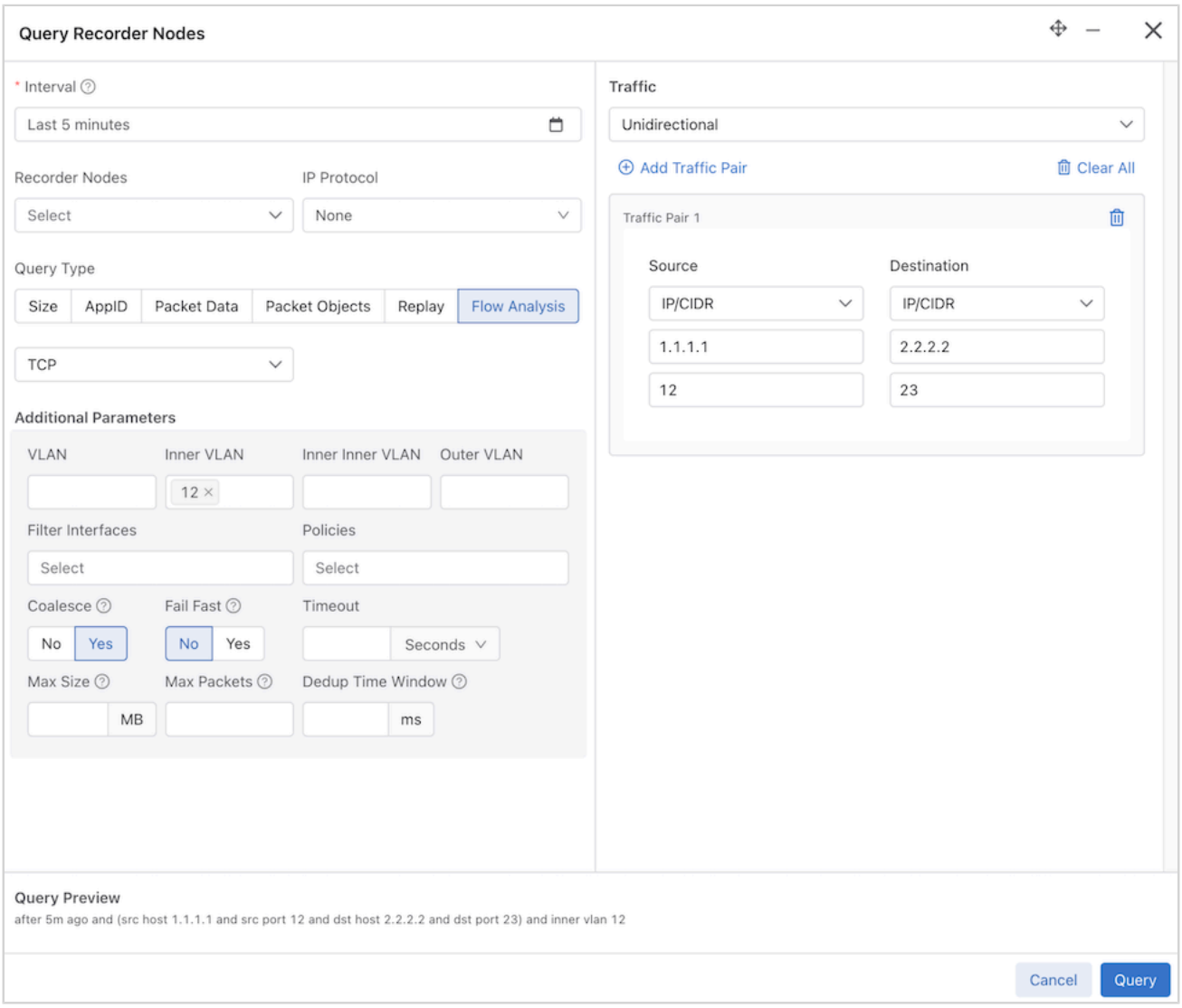
The following query actions are supported:

- **Interval**: Retrieve the oldest and most recent recorded packet timestamps. While hovering over the **info** icon provides information about the oldest and newest timestamp, to perform a query, you must enter the query time range using one of the following:
  - **Quick Windows**: A time range relative to the current time in which look for packets.
  - **Select Range**: A specific time range in which to look for packets.
- **Recorder Nodes**: Select a single or multiple Recorder Nodes from the drop-down list.
- **IP Protocol**: If required, select the IP protocol from the drop-down list or specify the numeric identifier of the protocol.
- **Size**: Provides the number of packets and their aggregate size in bytes that match the filter criteria specified.
- **AppID**: Performs deep packet inspection to identify applications communicating with the packets recorded and that match the filter criteria specified.
- **Packet Data**: Retrieves all the packets that match the filter criteria specified.
- **Packet Objects**: The packet object query extracts unencrypted HTTP objects from packets matching the given stenographer filter.
- **Replay**: Replays selected packets and transmits them to the specified delivery interface.

- **Flow Analysis**: Analyzes TCP flows for information such as maximum RTT, retransmissions, throughput, etc.
- **Traffic**:
  - **Any IP**: Include packets with the specified IP address in the IP header (either source or destination).
  - **Unidirectional**: Include packets with the specified source and/or destination IP address in the IP header.
- **Traffic Pair - Source Destination**:
  - **IP/CIDR or Mac**: Select packets with a specific source and destination IP or MAC address.
  - **Src Port**: Include packets with the specified protocol port number in the Src Port field in the IP header.
  - **Dst Port**: Include packets with the specified protocol port number in the Dst Port field in the IP header.
- **VLAN**: Select packets with a specific VLAN ID.
  - **Inner VLAN**
  - **Inner Inner VLAN**
  - **Outer VLAN**
- **Filter Interfaces**: Select the filter interfaces to restrict the query to those interfaces.
- **Policies**: Select the policies to restrict the query to those policies.
- **Coalesce**: Defines whether or not the data is coalesced if from multiple Recorder Nodes.
- **Fast Fail**: For multi-packet-recorder queries, if one packet recorder fails, fail the entire query immediately. Otherwise, continue obtaining a partial result from the remaining packet recorders.
- **Timeout**: Specify a timeout interval in **Seconds**, **Minutes**, or **Hours**.
- **Max Size**: This option is only available for packet queries. Specify the maximum number of bytes returned by a packet query in a PCAP file.
- **Max Packets**: This option is only available for packet queries. Specify the maximum number of packets returned by a packet query in a PCAP file.
- **Dedup time Window**: Refer to Deduplicate Packets for more information.

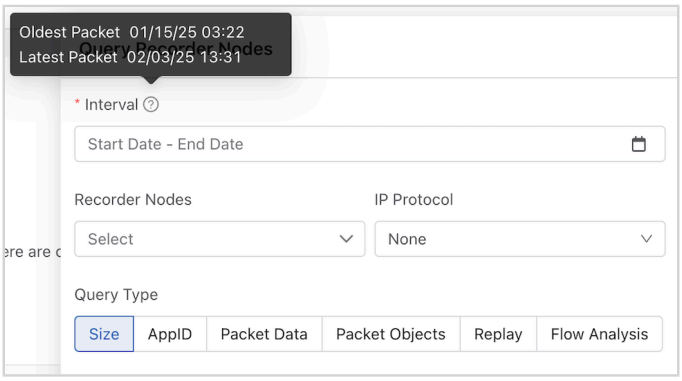
To query the **Recorder Nodes**, enter the required information. **Interval** details are mandatory, while other fields are optional. The current computed **Stenographer Query** string displays under **Query Preview**.

Figure 6-55: Query Recorder Nodes



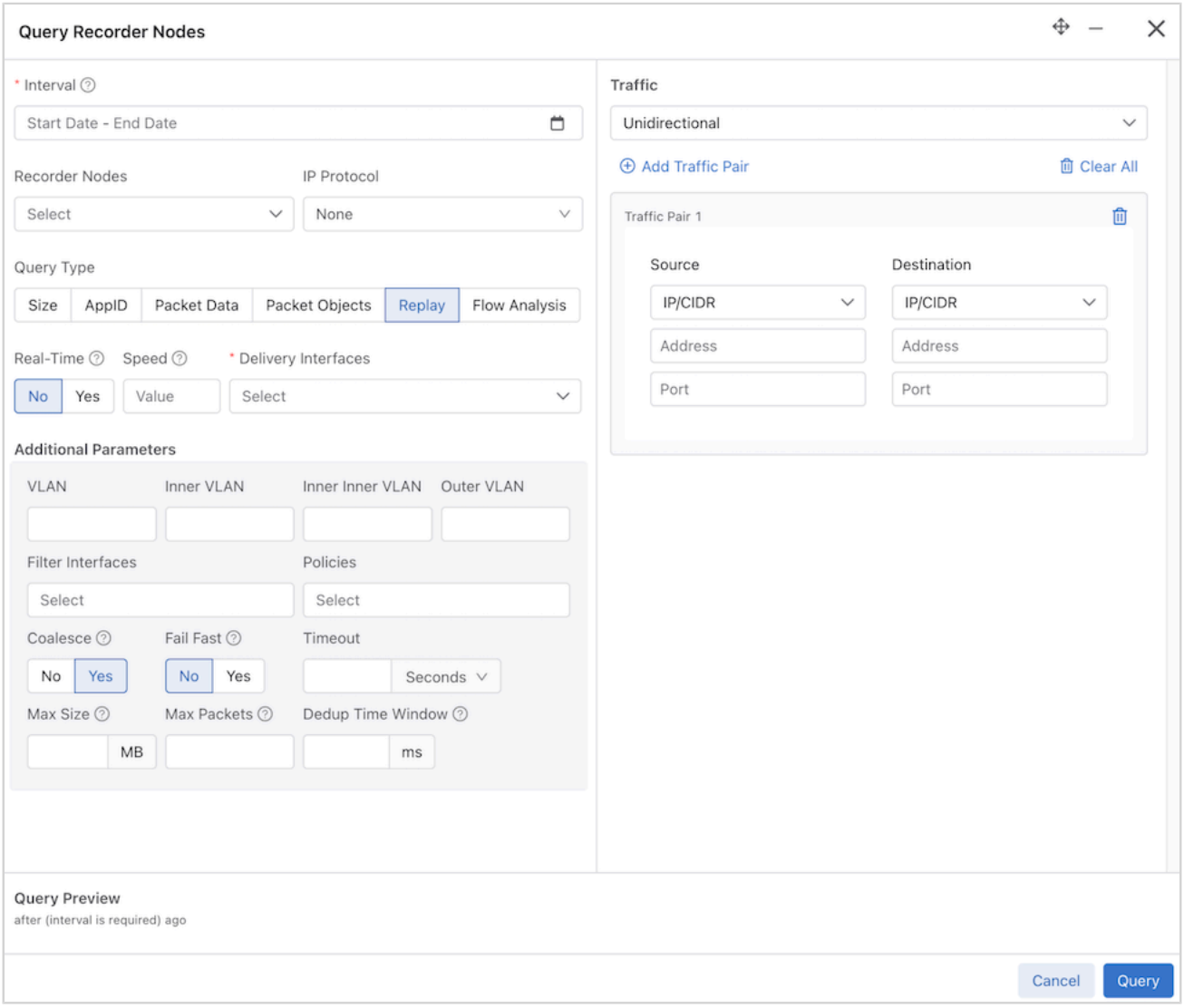
Hovering over the **info** icon for the **Interval** field displays the range of packets found.

Figure 6-56: Interval Details



The **Replay** query type has an additional mandatory field: **Delivery Interfaces**. Select **Delivery Interfaces** or **Delivery Interface Groups** from the drop-down.

Figure 6-57: Replay Fields

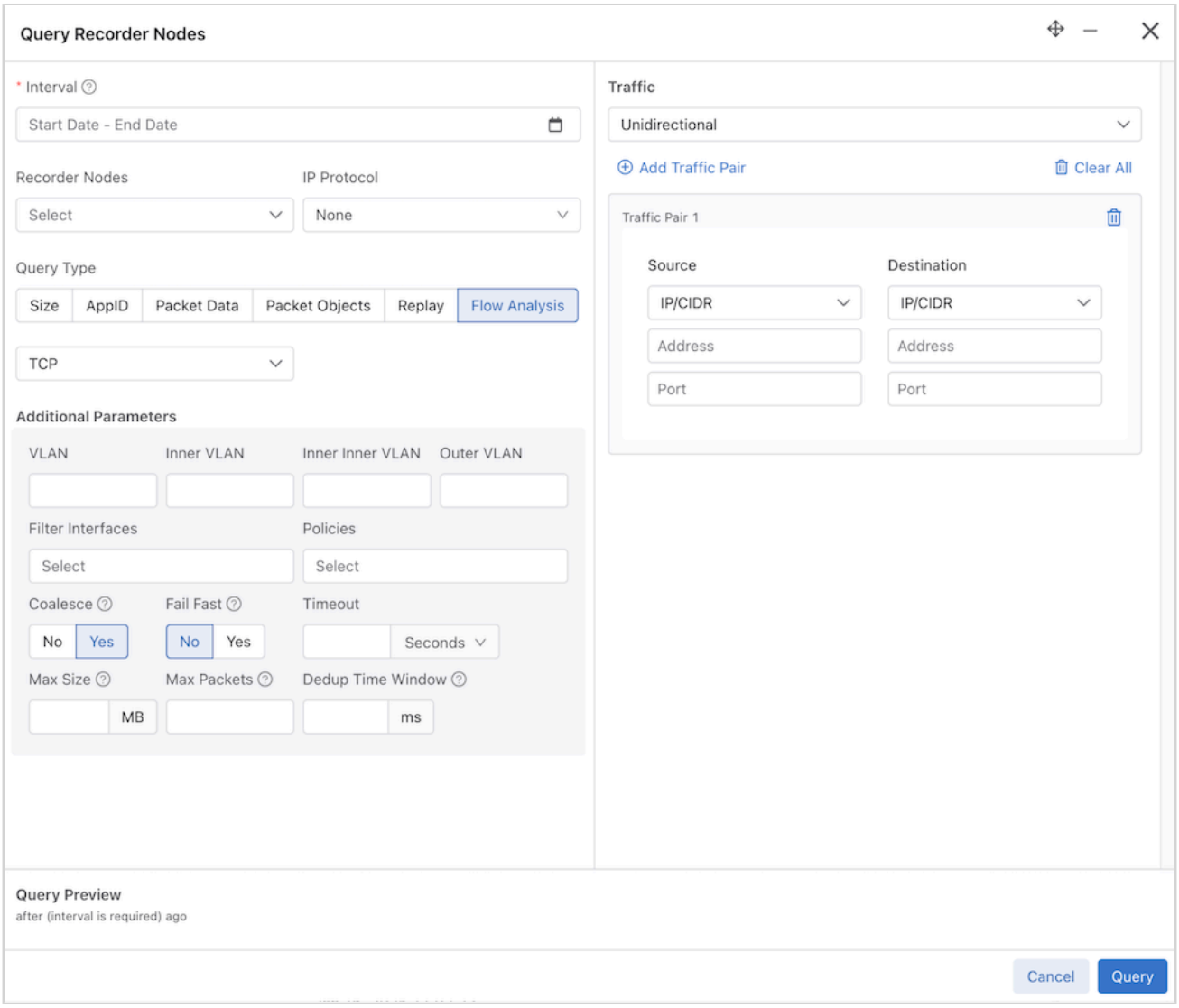


The **Flow Analysis** query type has an additional drop-down field of configuration parameters:

- **DNS**: Analyzes any DNS packets, extracting query and response metadata.
- **HTTP**: Analyzes HTTP packets, extracting request URLs, response codes, and statistics.
  - HTTP Request
  - HTTP Response
  - HTTP Stat
- **Hosts**: Identifies all the unique hosts that match the filter criteria specified.
- **IPv4**: Identifies and dissects distinct IPv4 flows.
- **IPv6**: Identifies and dissects distinct IPv6 flows.

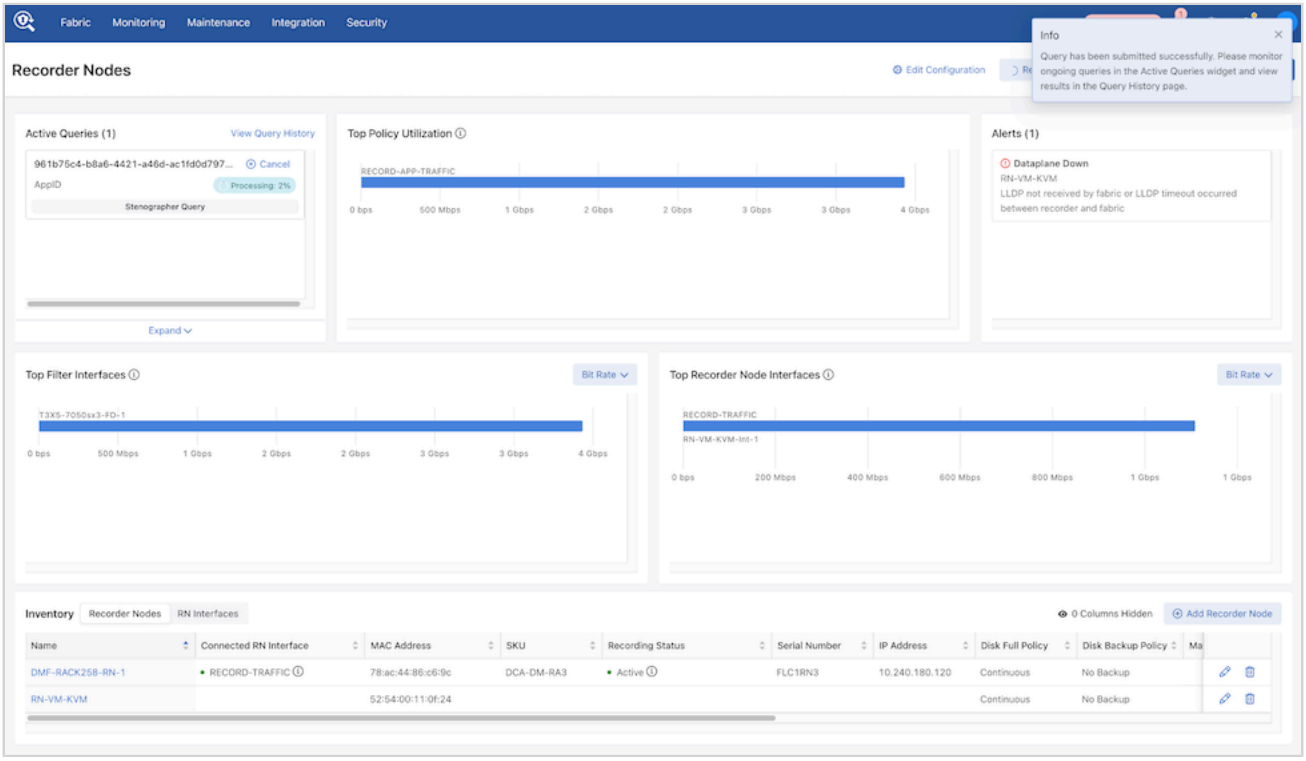
- **RTP Streams**: Characterizes the performance of Real Time Protocol streaming packets.
- **SIP Correlate**:
- **SIP Health**:
- **TCP**: Identifies and dissects distinct TCP flows.
- **TCP Flow Health**: Analyzes TCP flows for information such as maximum RTT, retransmissions, throughput, etc.
- **UDP**: Identifies and dissects distinct UDP flows.

Figure 6-58: Flow Analysis



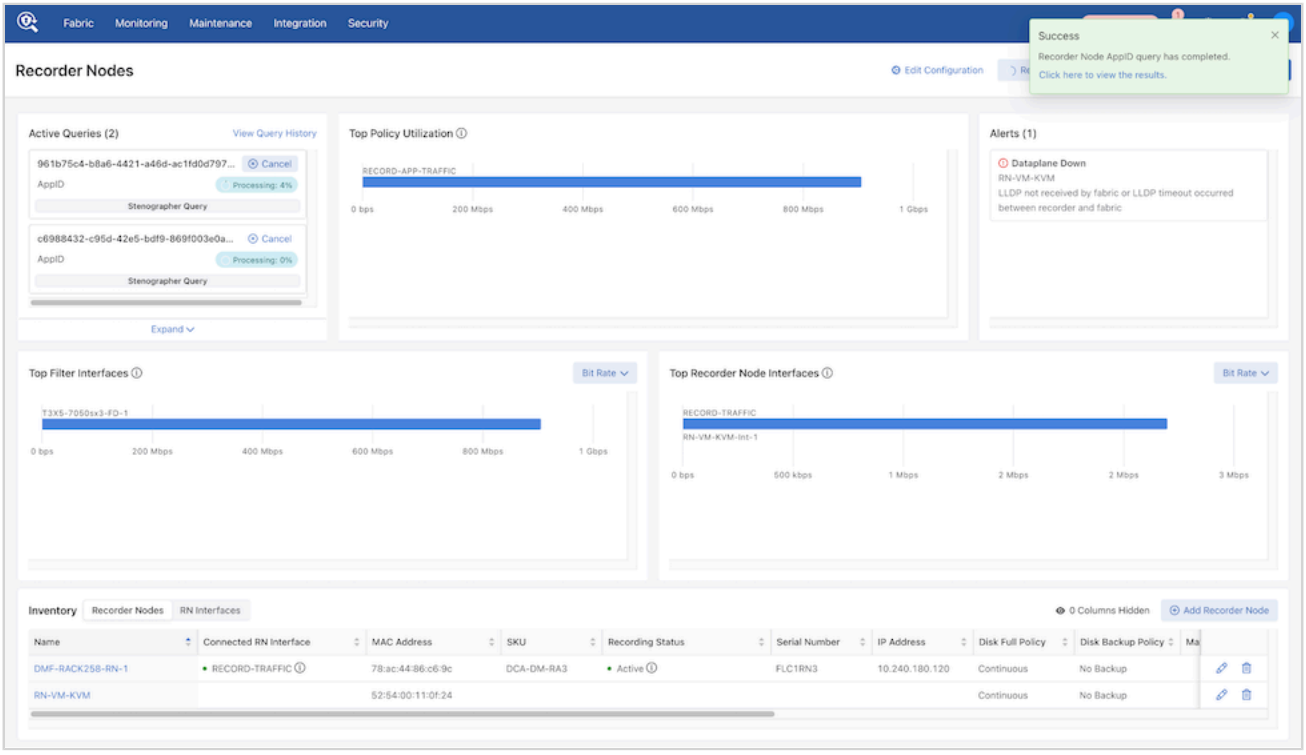
Upon entering the information and selecting **Query**, the window closes, and an info notification appears. The **Active Queries** are populated (unless the query completes quickly).

Figure 6-59: Info Notification



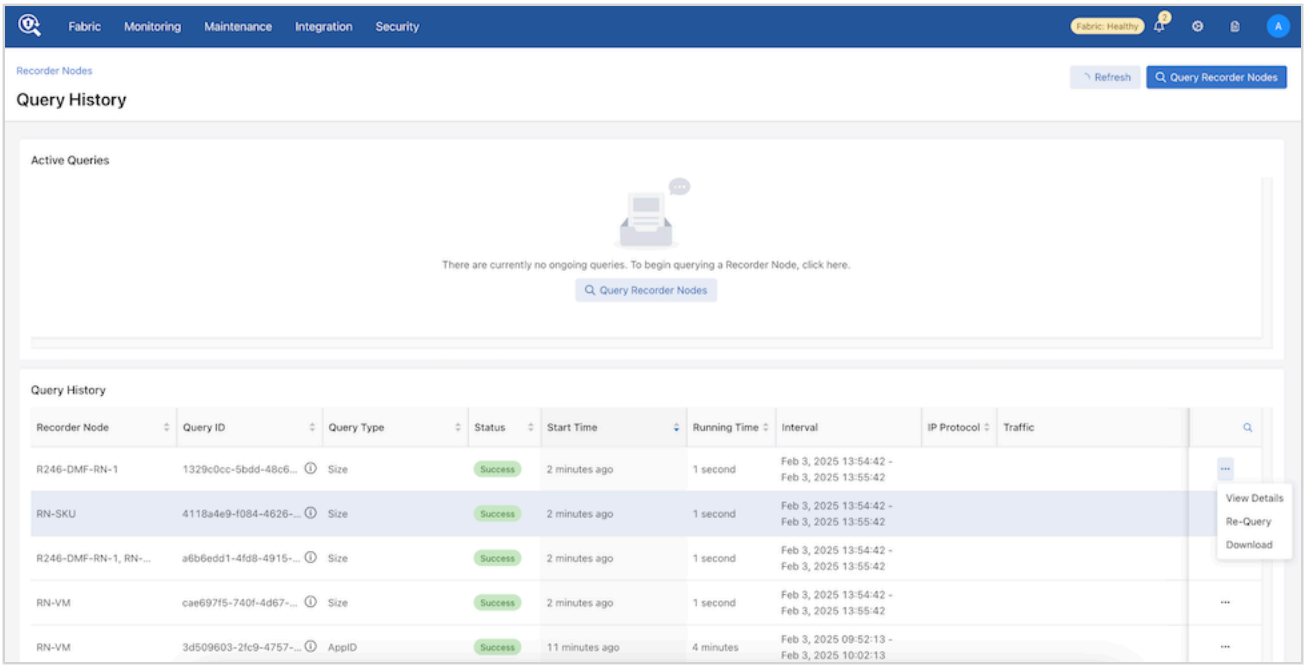
When the query completes, a success notification appears. Selecting the link goes to the **Query History** page.

Figure 6-60: Success Message



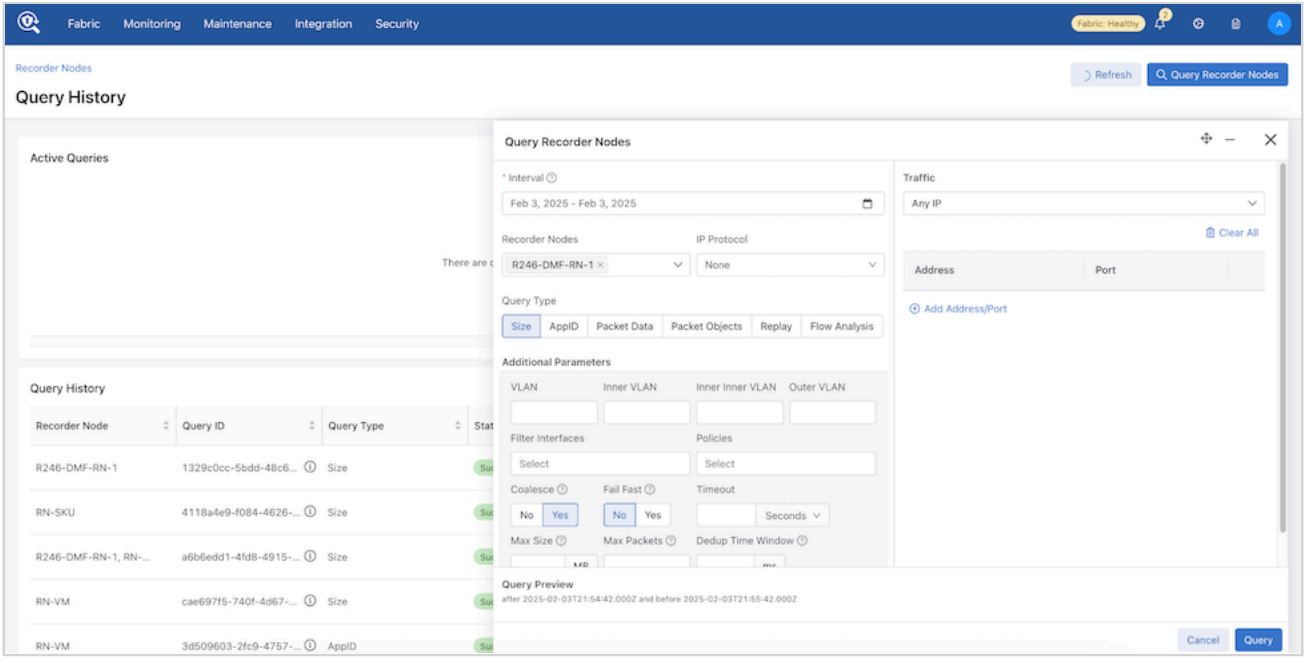
Select **Re-Query** from the **Query History** page to prepopulate certain querying fields.

Figure 6-61: Re-Query Option



Note: The system does not repopulate all fields because it does not retain all of a query's details. However, this entry point quickly modifies existing queries or speeds up commonly used queries.

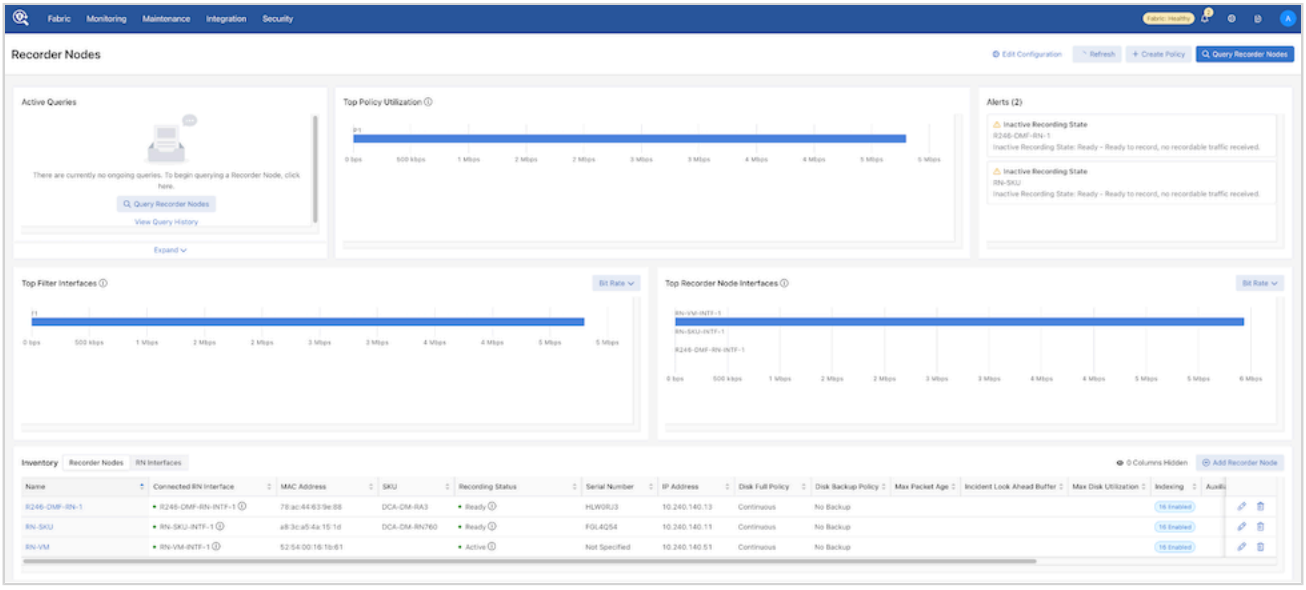
Figure 6-62: Re-Query Configuration



6.6.8.1 Viewing Query History

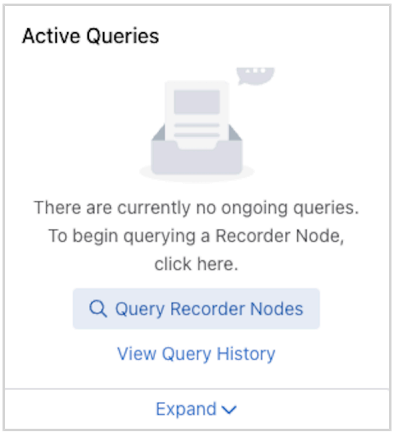
Navigate to **Monitoring** > **Recorder Nodes** and scroll down to the **Query History** section.

Figure 6-63: Recorder Nodes



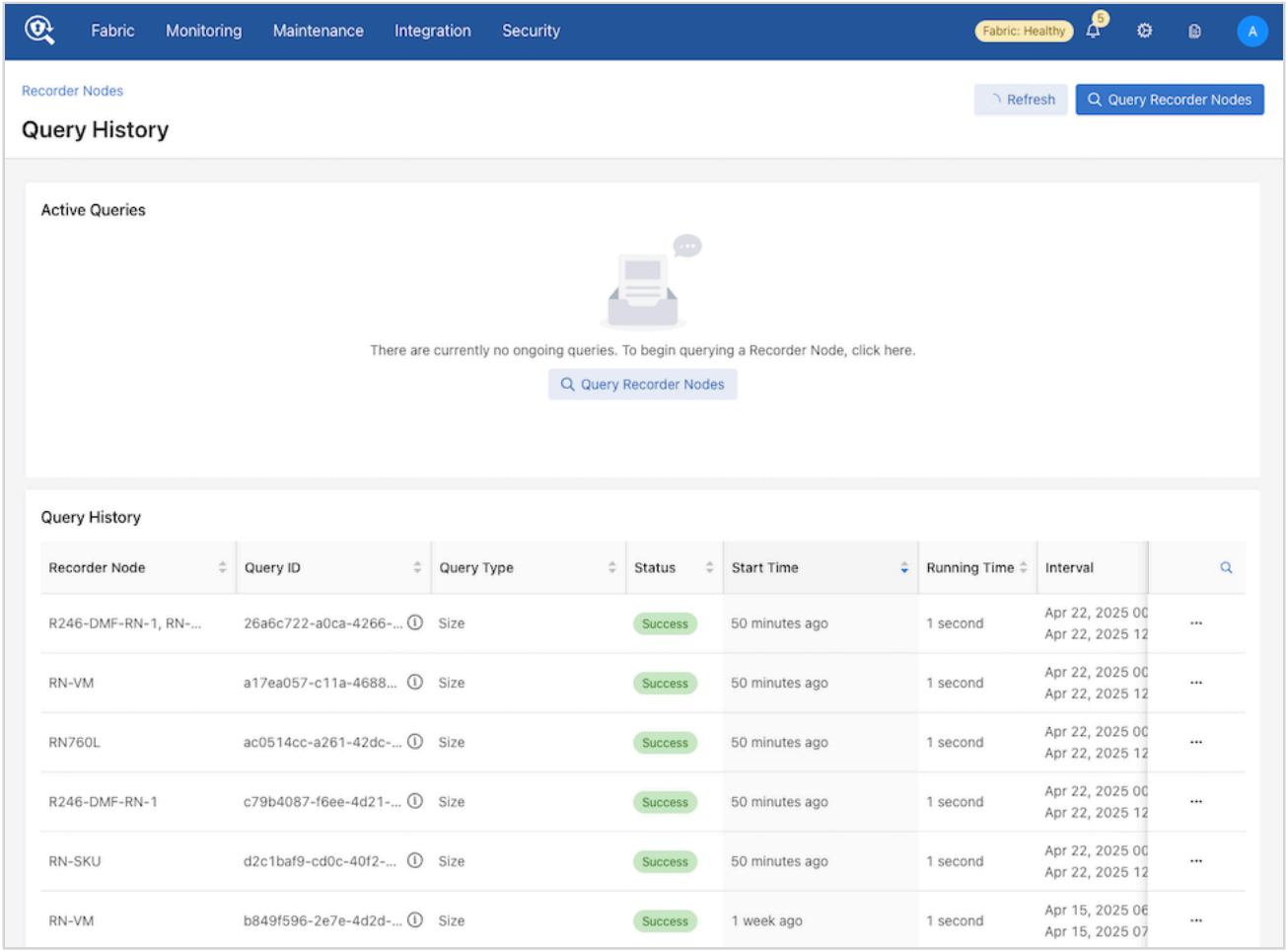
From **Active Queries**, select **View Query History**.

Figure 6-64: View Query History



The **Query History** dashboard appears.

Figure 6-65: Query Dashboard



While on the **Query History** page, use **Recorder Nodes** to return to the **Recorder Nodes** page.

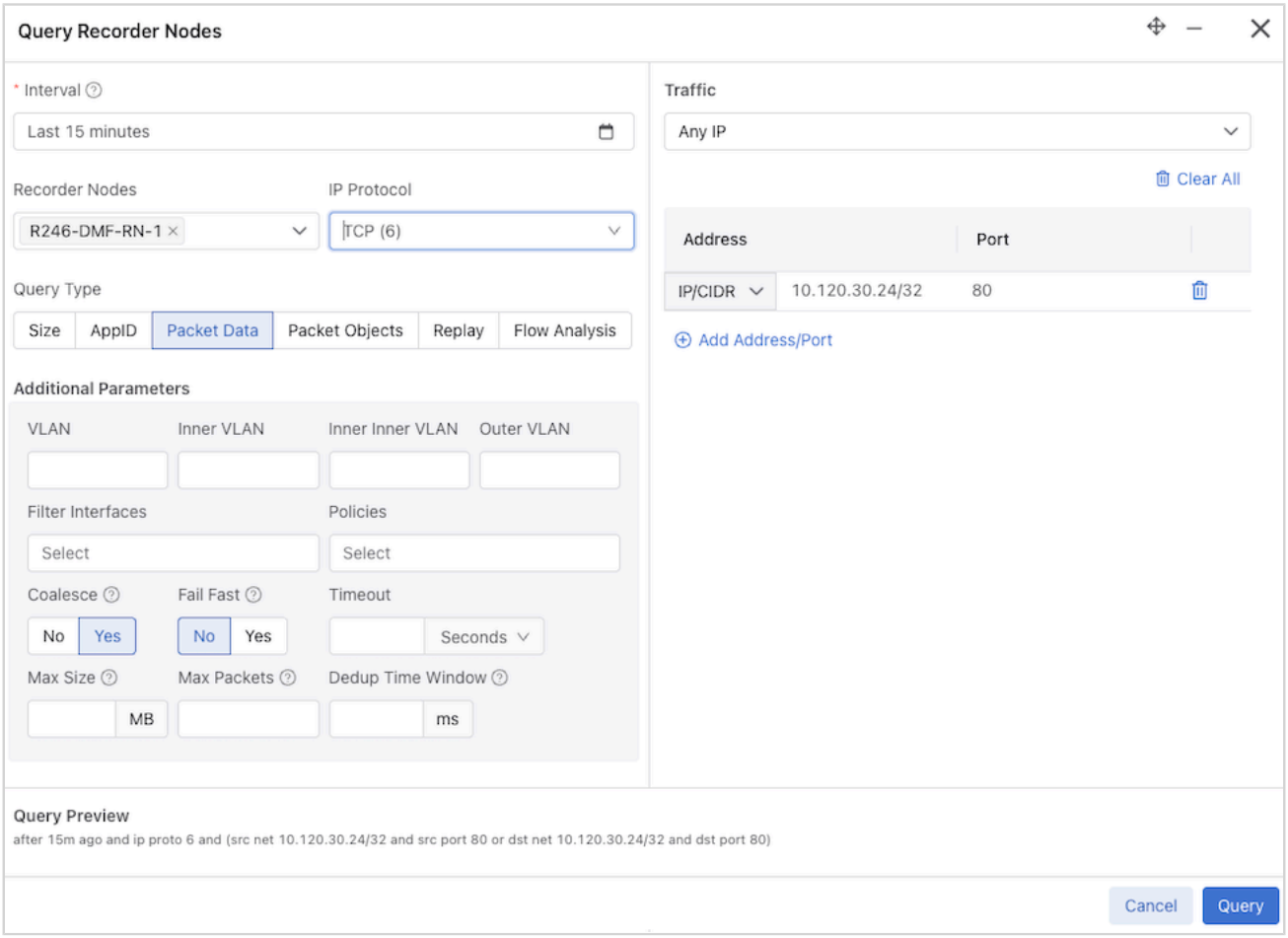
6.6.8.2 Multiple Queries

To run queries on recorded packets by the RN, navigate to the **Monitoring** > **Recorder Nodes** page.

Under the **Active Queries** section, select **Query Recorder Nodes** to select the type of analysis to run on the recorded packets.

After selecting the query type, use filters to limit or narrow the search to obtain specific results. Providing specific filters also helps to complete the query analysis faster. In the following example, the query result for the TCP query type will return the results for IP address ***10.240.30.24*** for the past ***15*** minutes.

Figure 6-66: Query Type



After entering the desired filters, select **Query**. Query status windows appear.

To view the results select **Active Queries**. If the query has finished, view the results using **Query History**.

Select **View Details** under the **More Options** button.

Figure 6-67: More Options

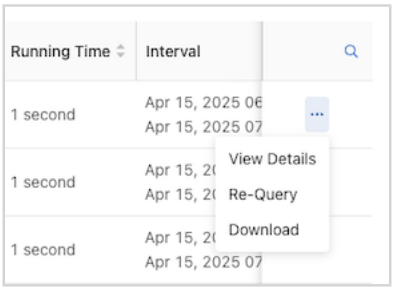
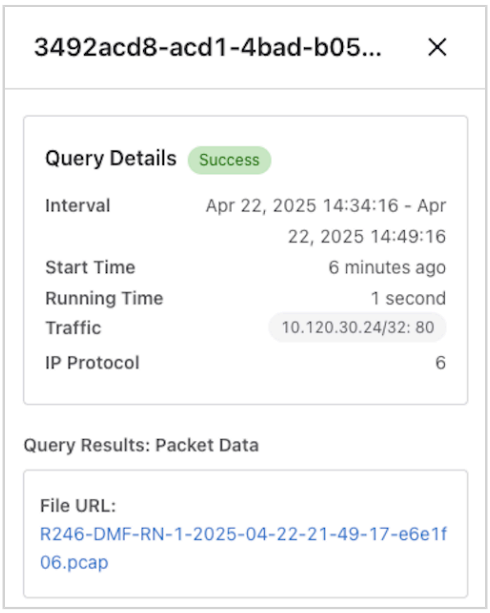


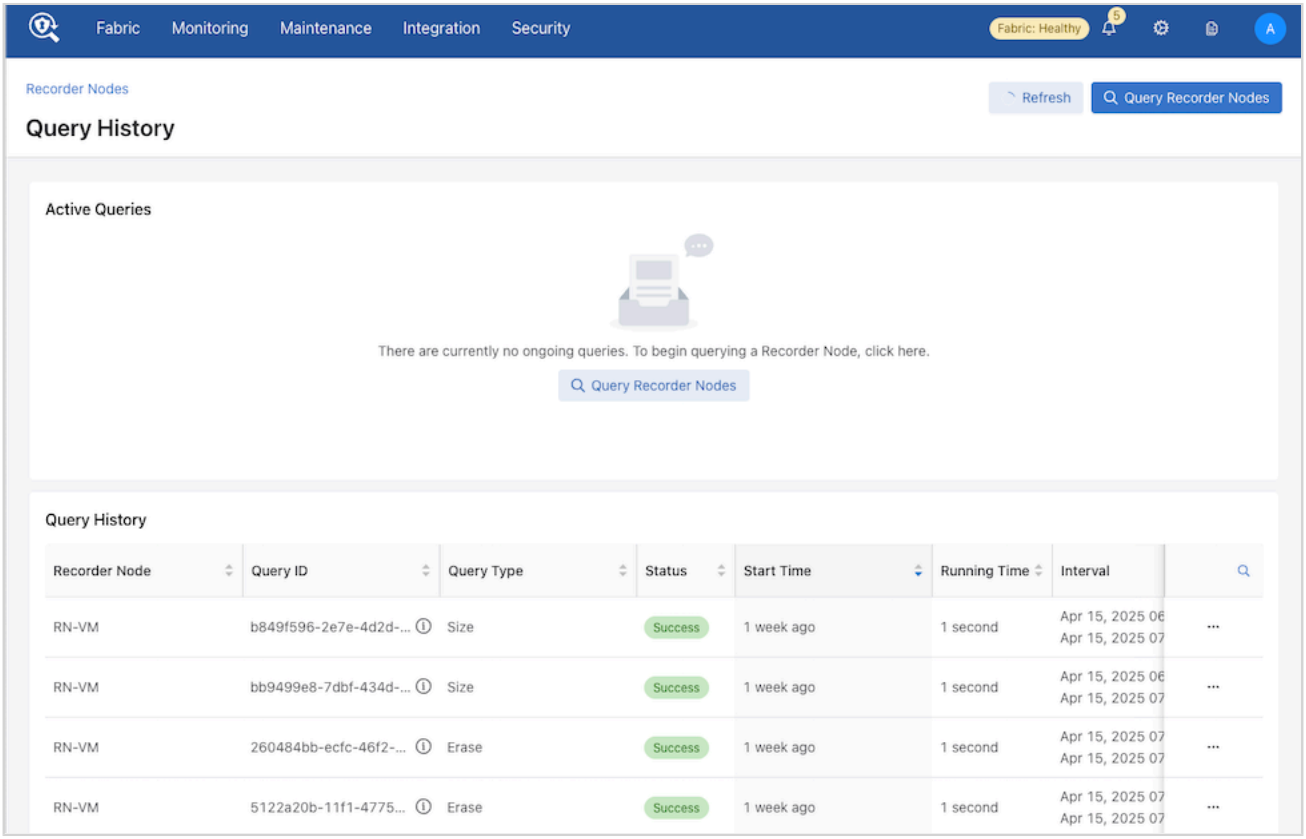
Figure 6-68: Query Details



6.6.8.3 Query Results

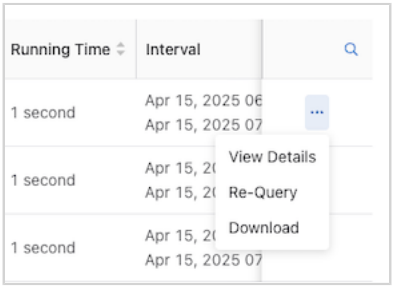
To view Recorder Node Query Results, navigate to **Query History**.

Figure 6-69: Query History



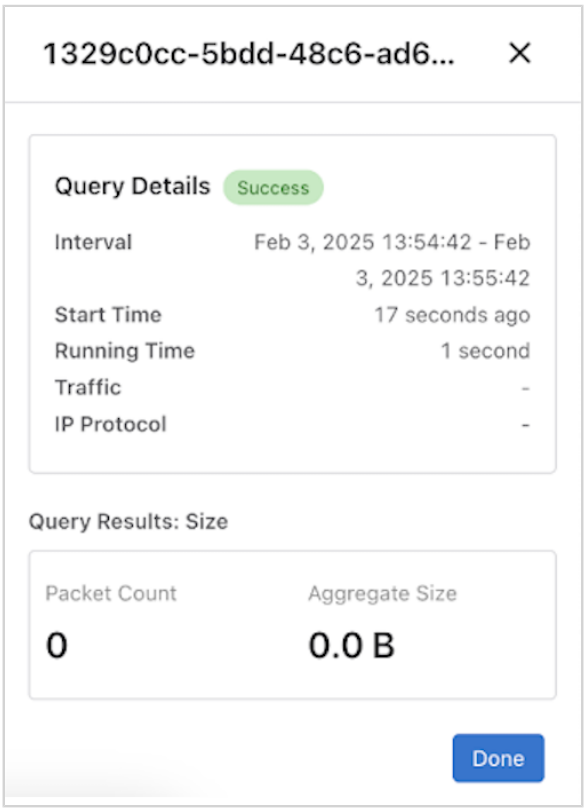
Select **View Details** under the **More Options** button.

Figure 6-70: More Options



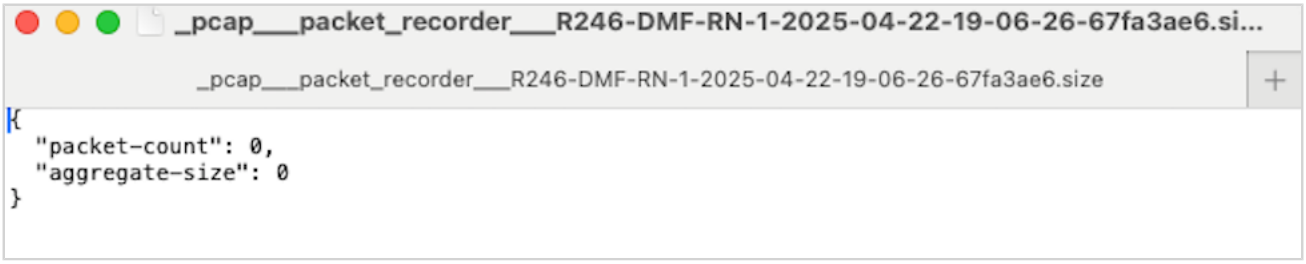
View Details displays the results of the query.

Figure 6-71: View Details



Selecting **Download** begins downloading the JSON data.

Figure 6-72: Downloaded Data

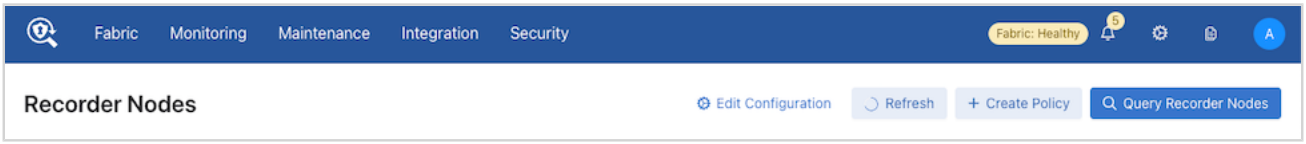


6.6.8.4 Deduplicate Packets

For Recorder Node queries, the recorded packets matching a specified query filter may contain duplicates when packet recording occurs at several different TAPs within the same network; i.e., as a packet moves through the network, it may be recorded multiple times. The dedup feature removes duplicate packets from the query results. By eliminating redundant information, packet deduplication improves query results' clarity, accuracy, and conciseness. Additionally, the dedup feature significantly reduces the size of query results obtained from packet query types.

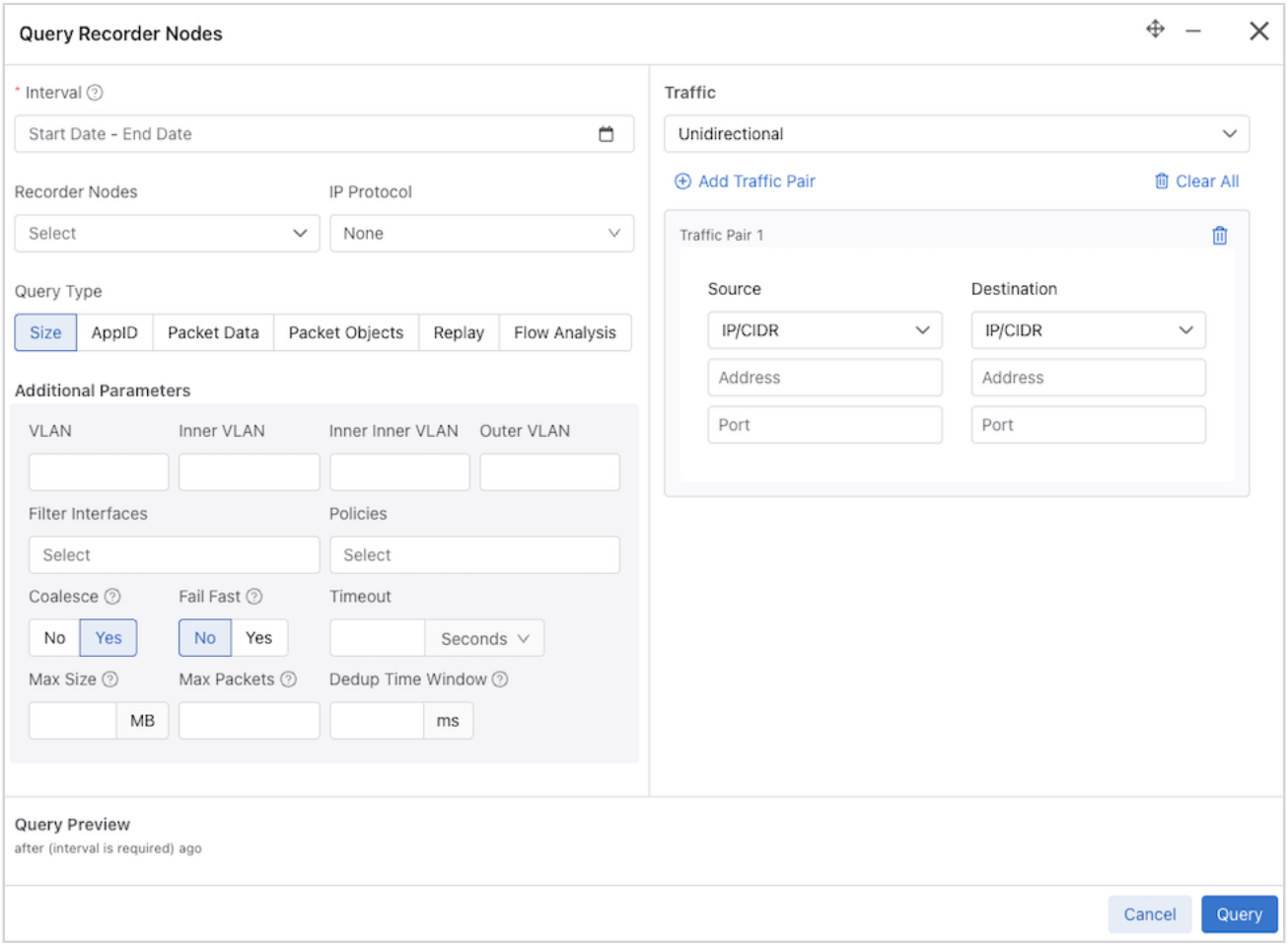
Navigate to **Monitoring** > **Recorder Nodes** > **Query Recorder Nodes**.

Figure 6-73: Query Recorder Nodes



The Query Recorder Nodes configuration window appears.

Figure 6-74: Query Recorder Nodes



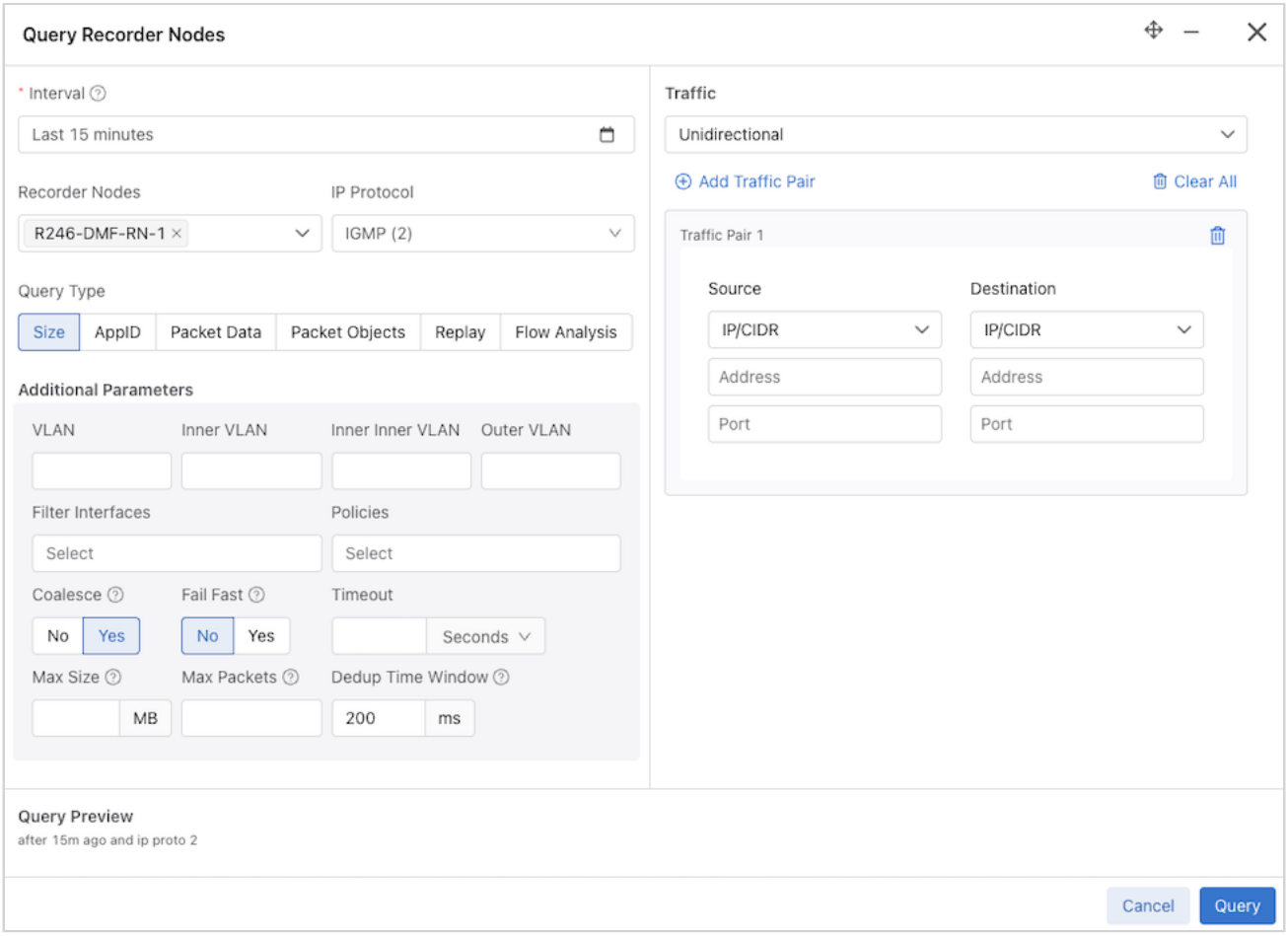
Deduplication is off by default for these queries. To enable deduplication, perform the following steps:

1. Choose a **Query Type**.

   Packet deduplication is available for the **Size**, **AppID**, **Packet Data**, **Packet Objects**, **Replay**, and **Flow Analysis** query types.
2. Specify a time window (in milliseconds) by entering an integer between 0 and 999 (inclusive) in the **Dedup Time Window** field.
3. Select **Query** to continue.

The following example illustrates enabling deduplication for a **Size** query specifying a **Dedup Time Window** value of **200 ms**.

Figure 6-75: Dedup Parameters



6.6.9 Manage Access to the Recorder Node

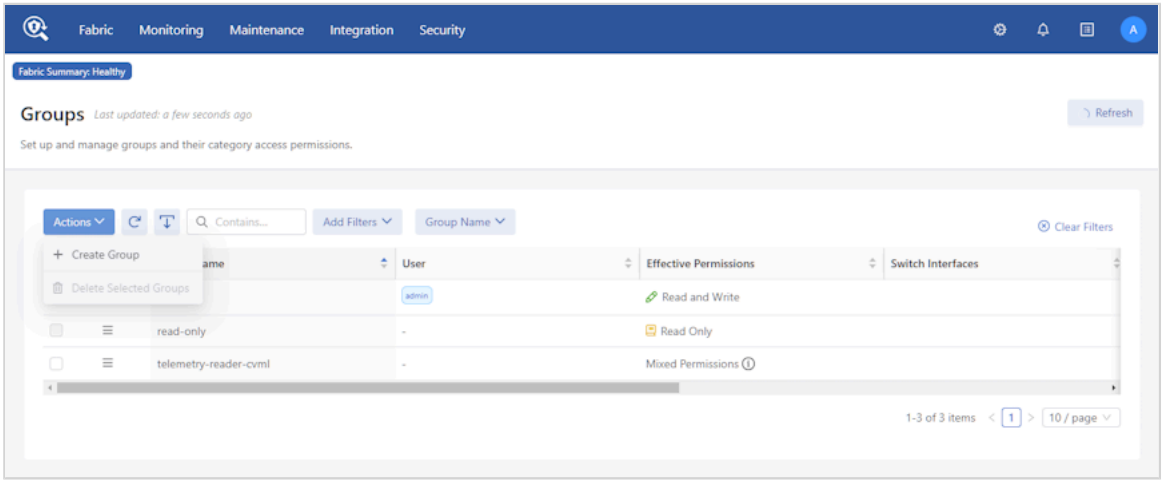
Use Role-Based Access Control (RBAC) to manage access to the DANZ Monitoring Fabric (DMF) Recorder Node (RN) by associating the RN with an RBAC group.

To restrict access for a specific RN to a specific RBAC group, use the following instructions.

6.6.9.1 RBAC Configuration

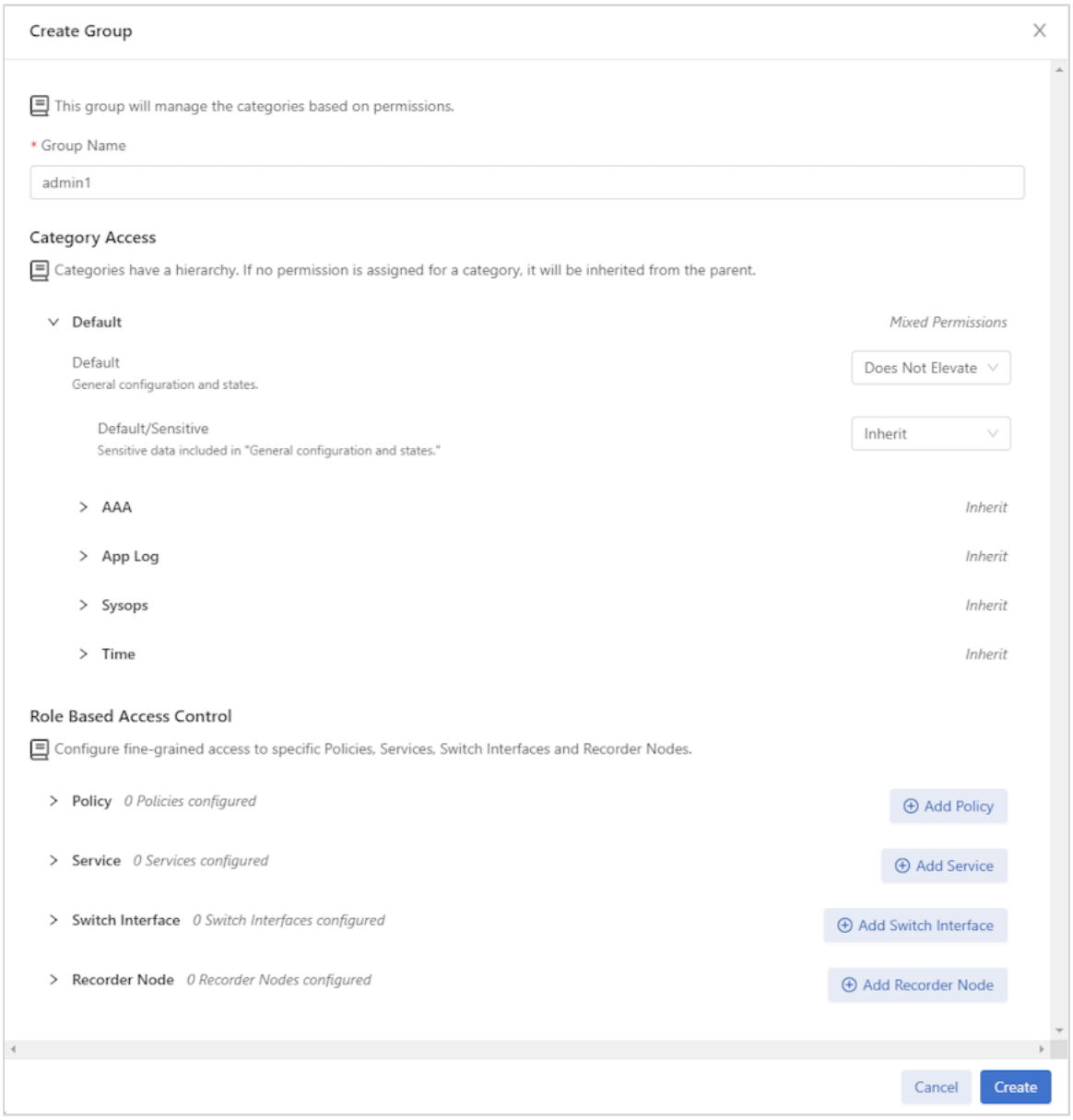
1. Select **Security > Groups**, and select **Edit** from the **Actions** and select **+ Create Group**.

Figure 6-76: Create Security Group



2. Enter a **Group Name**.

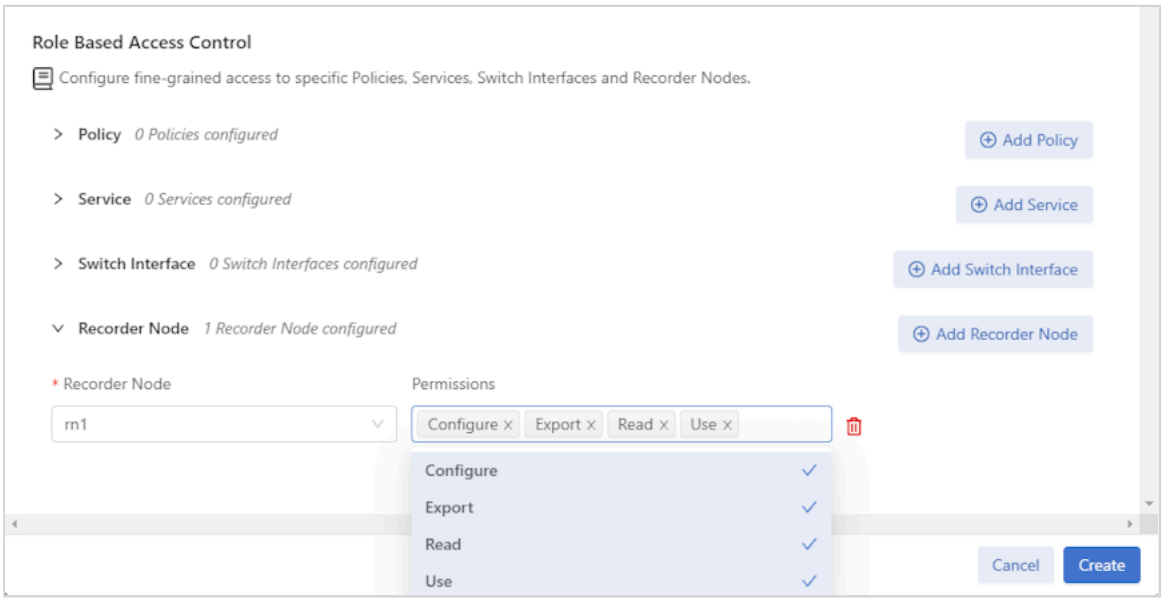
Figure 6-77: Create Group



3. Under the **Role Based Access Control** section select **Add Recorder Node**.

4. Select the **Recorder Node** from the selection list, and assign the permissions required.
   - **Read**: The user can view recorded packets.
   - **Use**: The user can define and run queries.
   - **Configure**: The user can configure packet recorder instances and interfaces.
   - **Export**: The user can export packets to a different device.

Figure 6-78: Associate Recorder Node



5. Select **Create**.

6.6.9.2 Rename a RBAC Group

This topic describes the workflow for renaming a **Group Name** in DMF.

Overview

Navigate to **Security** > **Groups** and select **Groups**.

Figure 6-79: Security Groups

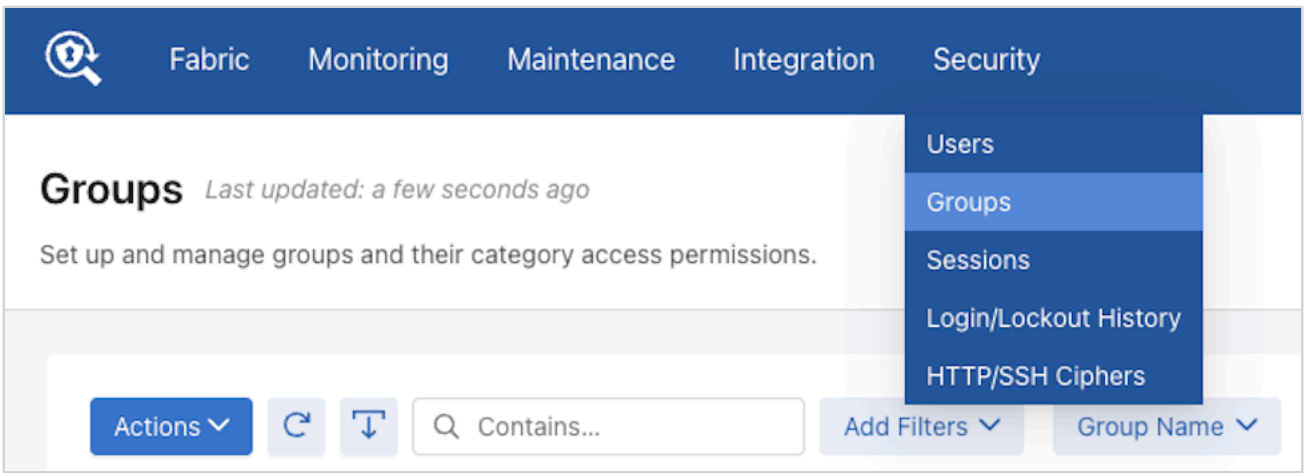
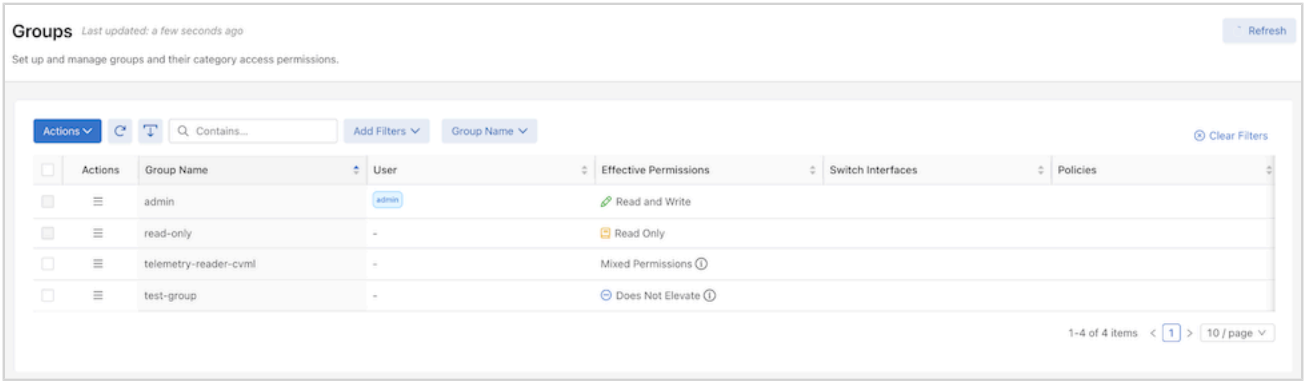


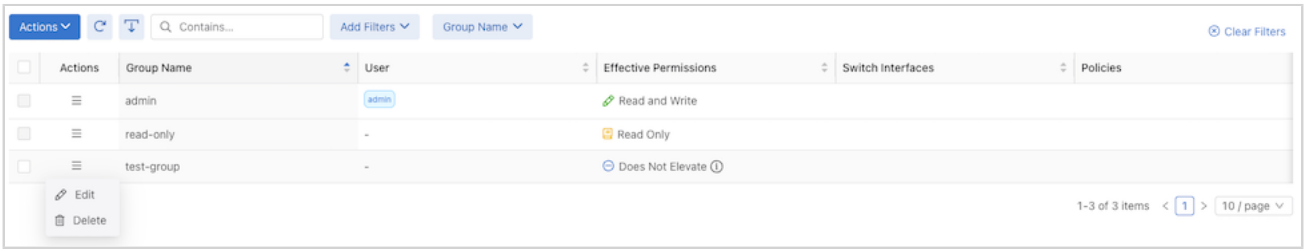
Figure 6-80: Groups Dashboard



Renaming a Group Name

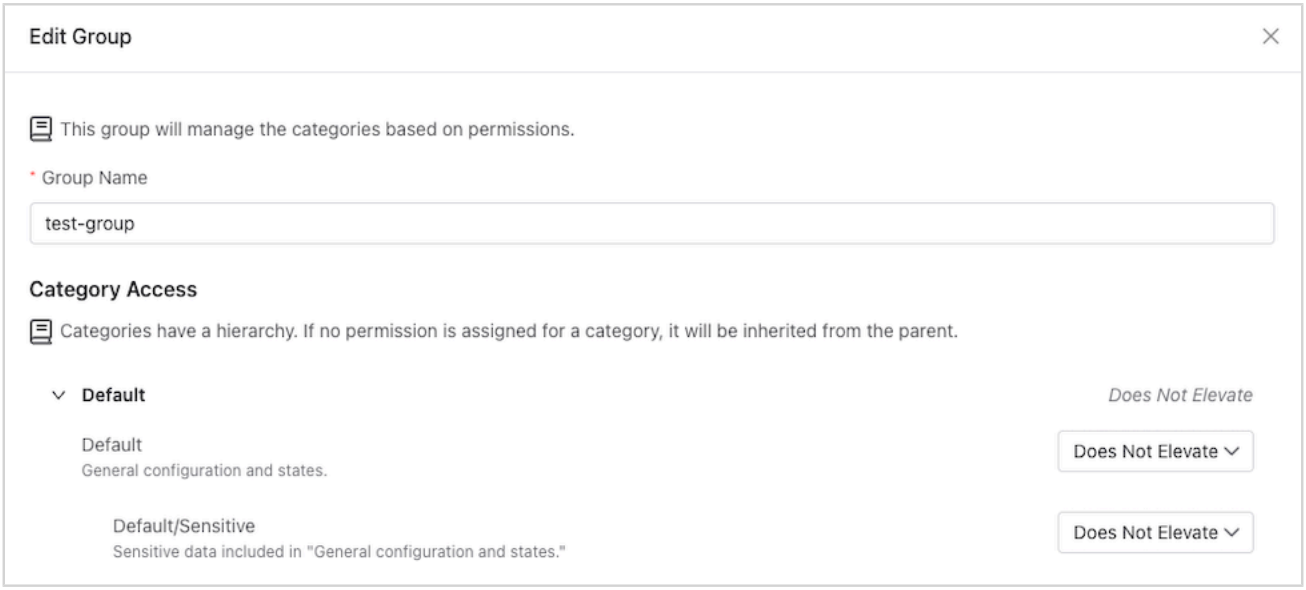
To update a **Group Name**, such as changing **`test-group`** to **`test-group-updated`**, select **Edit** in the row menu action.

Figure 6-81: Edit



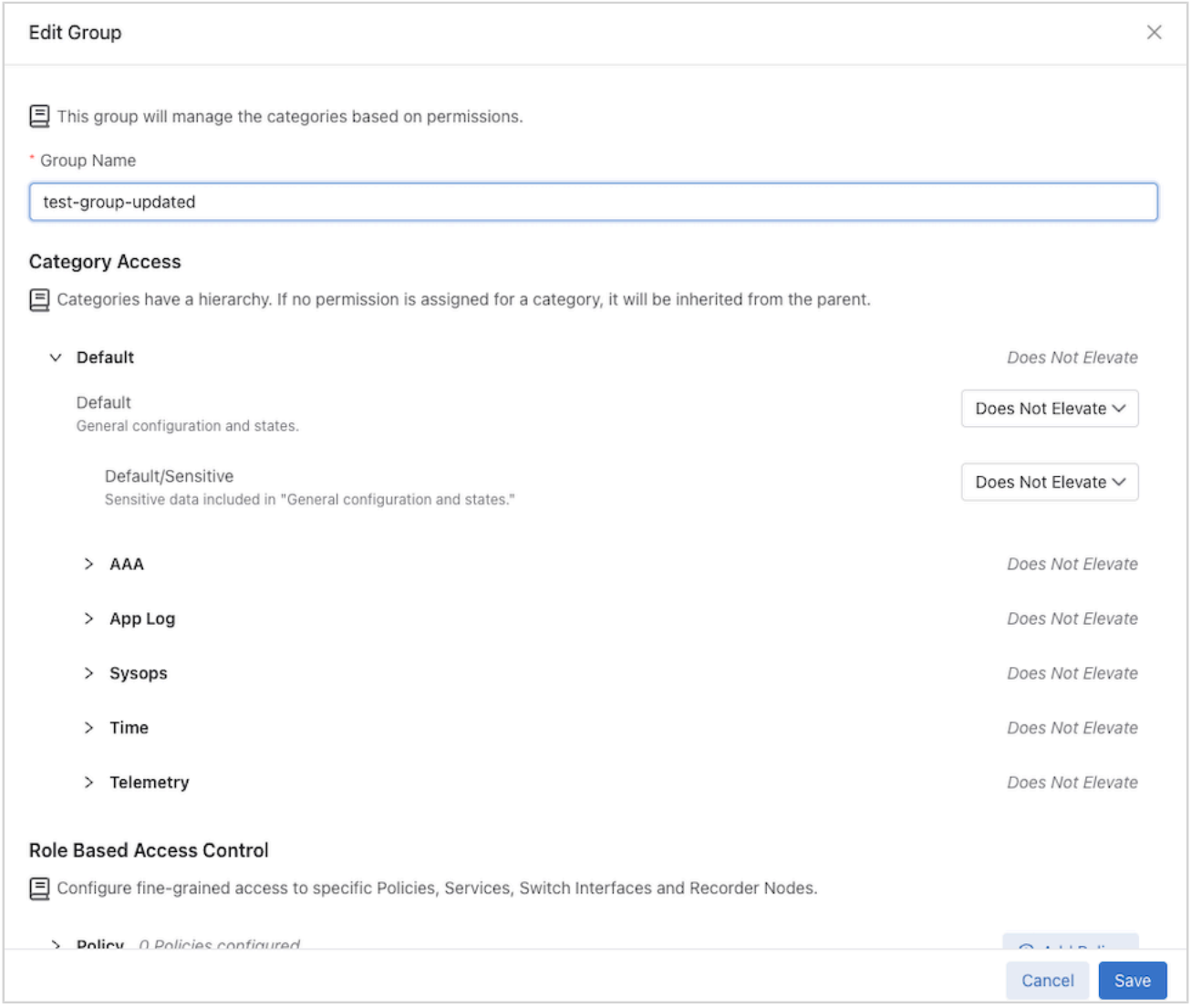
An **Edit Group** window displays.

Figure 6-82: Edit Group



Enter the new **Group Name**.

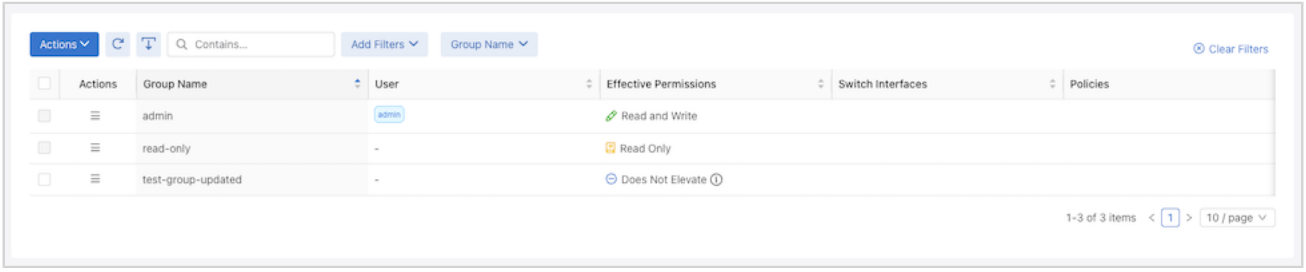
Figure 6-83: Updated Group Name



Select **Save** to apply the change.

DMF updates the **Group Name**.

Figure 6-84: Updated Group Name



6.6.10 Enabling Egress sFlow® on Recorder Node Interfaces

Enable egress sFlow®* to sample traffic sent to any DANZ Monitoring Fabric (DMF) Recorder Node (RN) attached to the fabric. Examining these sampled packets on a configured sFlow collector allows the identification of post-match-rule flows recorded by the RNs without performing a query against the RNs. While not explicitly required, Arista Networks highly recommends using the DMF Analytics Node (AN) as the configured sFlow collector, as it can automatically identify packets sampled utilizing this feature.

Platform Compatibility

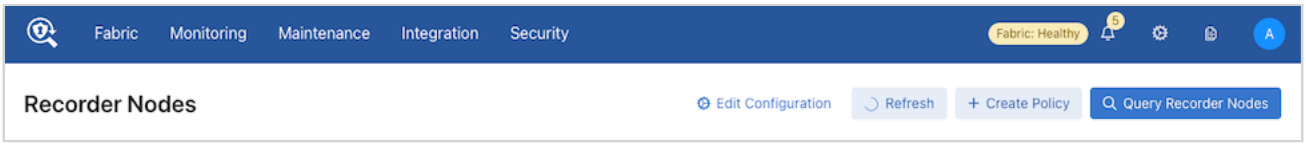
All platforms apart from the following series:

- DCS-7280R
- DCS-7280R2
- DCS-7500R
- DCS-7020
- DCS-7050X4

Configuration

After configuring the fabric for sFlow and setting up the sFlow collector, navigate to **Monitoring** > **Recorder Nodes**.

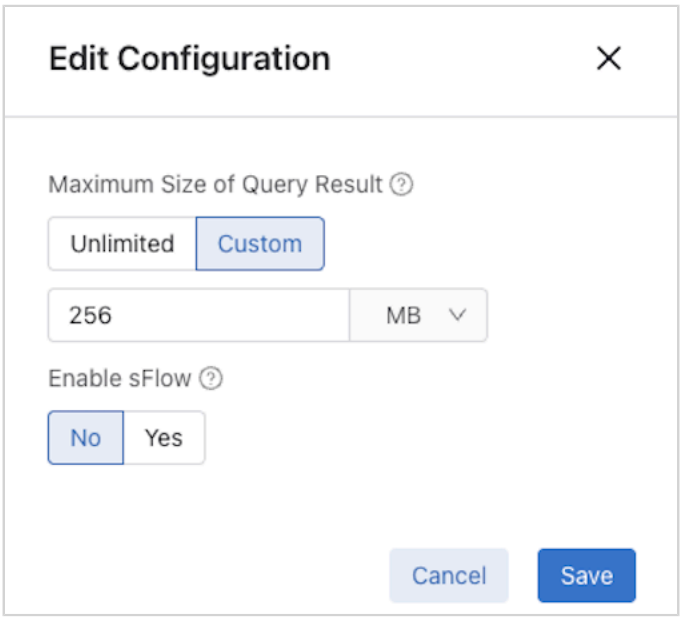
Figure 6-85: Query Recorder Nodes



* sFlow® is a registered trademark of Inmon Corp.

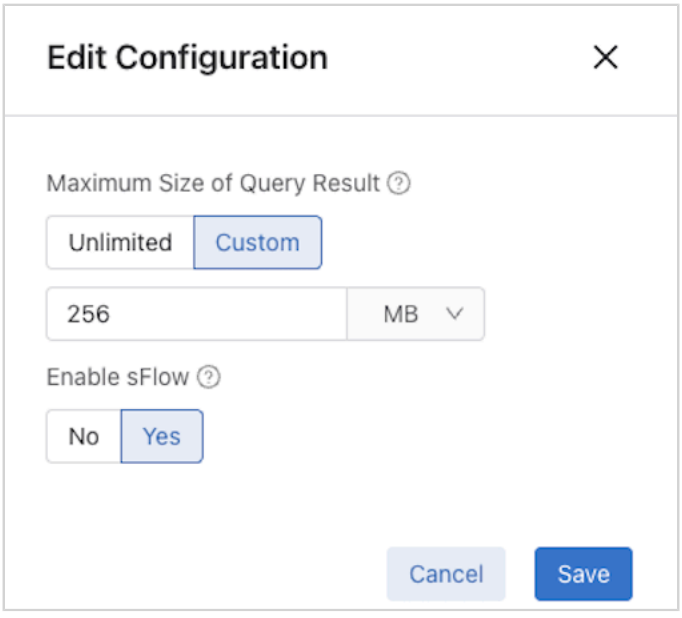
Select **Edit Configuration** and the configuration menu appears.

Figure 6-86: sFlow Configuration



Set **Enable sFlow** to **Yes**.

Figure 6-87: sFlow Enabled



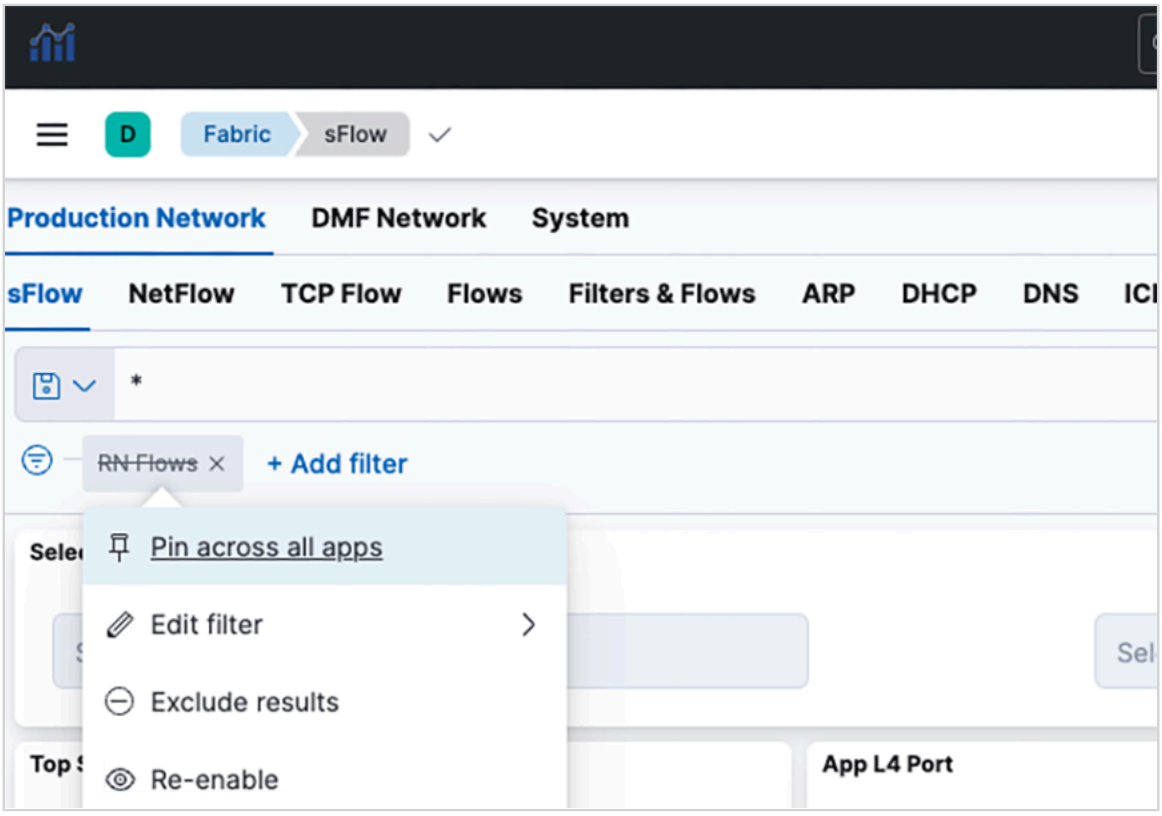
Select **Save**.

DMF Analytics Node

When using a DMF Analytics Node as the sFlow collector, it has a dashboard to display the results from this feature. To access the results:

1. Navigate to the **sFlow** dashboard from the **Fabric** dashboard.
2. Select the disabled **RN Flows** filter.
3. Select the option to **Re-enable** the filter.

Figure 6-88: Re-enable sFlow



6.6.10.1 Troubleshooting Egress sFlow Configurations

Switches not affiliated with a sFlow collector (either a global sFlow collector or a switch-specific sFlow collector) do not have an active feature even if the feature is enabled. Ensure the fabric is set up for sFlow and a configured sFlow collector exists. To verify that a configured global sFlow collector exists, use the command:

```
Controller-1# show sflow default
```

A configured collector appears as an entry in the table under the column labeled **collector**. Alternatively, to verify a configured collector exists for a given switch, use the command:

```
Controller-1#  show switch switch-name table sflow-collector
```

This command displays a table with one entry per configured collector.

A ***feature-unsupported-on-device*** warning appears when connecting an unsupported switch to an RN. The feature does not sample packets passing to an RN from an unsupported switch. View any such warnings using the GUI or using the following CLI command:

```
Controller-1#  show fabric warnings feature-unsupported-on-device
```

To verify the feature is active on a given switch, use the command:

```
Controller-1#  show switch switch-name table sflow-sample
```

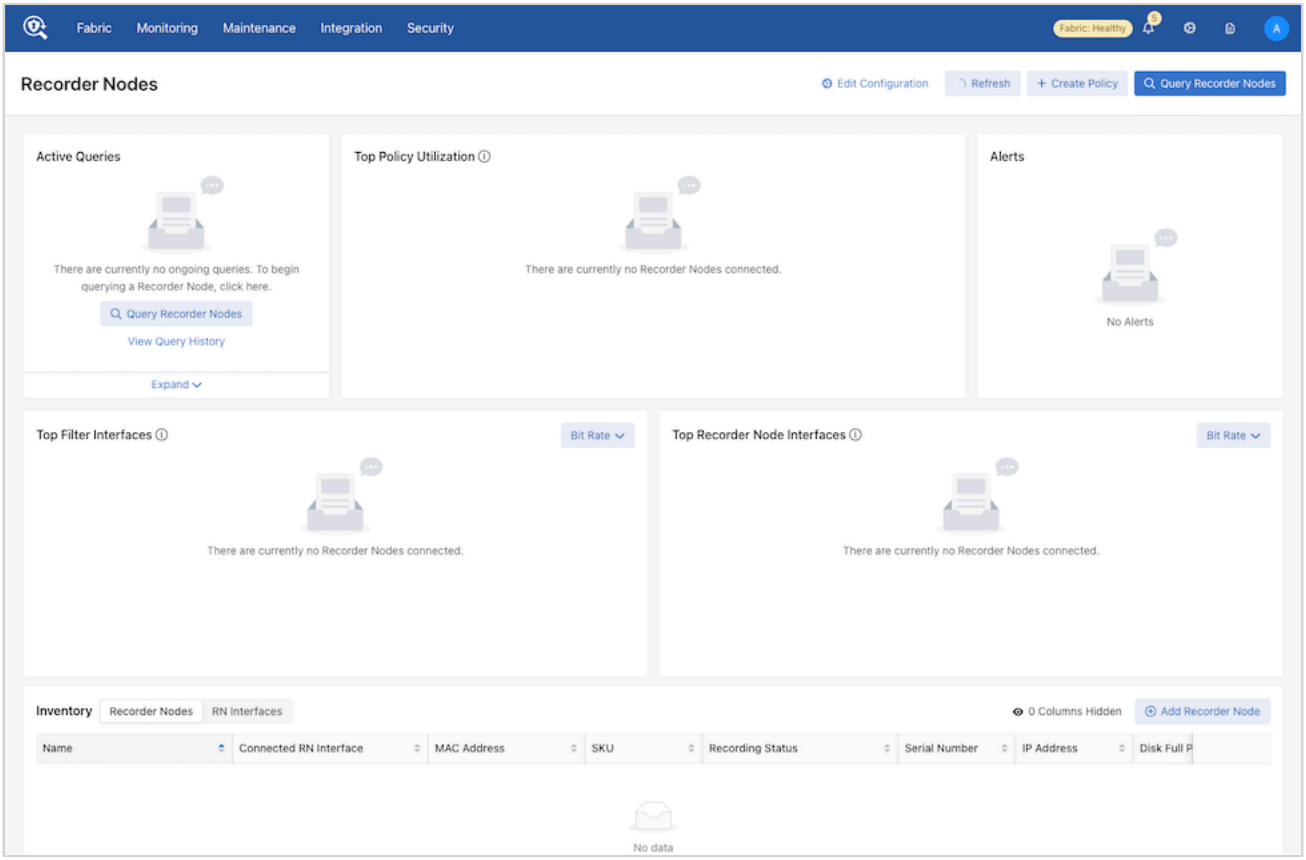
If the feature is enabled, the entry values associated with the ports connected to an RN would include an `EgressSamplingRate(number)` with a `number` greater than `0`. The following example illustrates `Port(1)` on *`switch-name`* connecting to an RN.

```
Controller-1# show switch <switch-name> table sflow-sample
#  Sflow-sample Device name    Entry key Entry value
--|------------|--------------|---------|-------------------------------------------------------------------------------------|
53  52          <switch-name>  Port(1)   SamplingRate(0), EgressSamplingRate(10000), HeaderSize(128), Interval(10000)
```

6.6.11 Empty State

Users installing DMF 8.7.0 for the first-time who have not migrated from a previous DMF release will see an empty-state **Recorder Nodes** dashboard, as shown in the following example.

Figure 6-89: Recorder Node - Empty State



Please refer to the *DMF Deployment Guide* for more information on installing and configuring a DMF Recorder Node.

6.7 Using the Recorder Node Command Line Interface

6.7.1 Manage the DMF Recorder Node

6.7.1.1 Basic Configuration

To perform basic Recorder Node (RN) configuration, perform the following steps:

1. Assign a name to the RN device.

```
controller-1(config)# recorder-node device rn-alias
```

2. Set the MAC address of the RN.

```
controller-1(config-recorder-node)# mac 18:66:da:fb:6d:b4
```

Determine from the chassis ID of connected devices if the management MAC is unknown.

```
controller-1> show connected-devices packet-recorder
# Switch   If Name    DMF name     SPAN? Device Name  Device Description                    Chassis ID        Port ID           Port Description  Management Address Protocol
-|--------|----------|------------|-----|-----------|-------------------------------------|-----------------|-----------------|-----------------|------------------|--------|
1 bt-1b9-1 ethernet50             False recorder-1   DMF Recorder Node, SN MLEYHM2         18:66:da:fb:6d:b4  3c:fd:fe:1f:0f:82  enp130s0f1        10.4.100.200       LLDP
```

3. Define the RN interface name.

```
controller-1(config)# recorder-fabric interface Intf-alias
controller-1(config-pkt-rec-intf)#
```

Assign any alphanumeric identifier for the recorder node interface name, which changes the submode to ***config-pkt-rec-intf***, to provide an optional description. This submode allows specifying the switch and interface where the RN is connected.

4. Provide an optional description and identify the switch interface connected to the RN.

```
controller-1(config-pkt-rec-intf)# description 'Delivery point for recorder-node'
controller-1(config-pkt-rec-intf)# recorder-interface switch Switch-z9100 ethernet37
```

5. (Optional) **Recording**: Enabled by default. To disable recording, enter the following commands:

```
controller-1(config)# recorder-node device rn-alias
controller-1(config-recorder-node)# no record
```

6. (Optional) **Disk Full Policy**: By default, **Disk Full Policy** is set to **`rolling-fifo`**, deleting the oldest packets to make room for newer packets when RN disks are full. This configuration can be changed to **`stop-and-wait`**, allowing the RN to stop recording until disk space becomes available. Enter the commands below to configure **Disk Full Policy** to **`stop-and-wait`**.

```
controller-1(config)# recorder-node device rn-alias
controller-1(config-recorder-node)# when-disk-full stop-and-wait
```

7. **Backup Disk Policy**: Define the backup disk policy to select the secondary volume and select one of the following three options:

```
controller-1(config-recorder-node)# backup-volume
local-fallback Set local disk as backup when remote disk is unreachable
no-backup      Do not use any backup volume (default selection)
remote-extend  Set remote volume to extend local main disk
```

The **`no-backup`** mode is the default mode. The other two modes require that the Recorder Node have a set of recording disks and a connection to an Isilon cluster mounted via NFS. Configure this remote storage from the DMF Controller.

8. (Optional) **Max Packet Age**: This defines the maximum age in minutes of any packet in the RN. By default, **Max Packet Age** is unset, which means no limit is enforced. When setting a **Max Packet Age**, packets recorded on the RN are discarded after the minutes specified. To set the maximum number of minutes that recorded packets will be kept on the RN, enter the following commands:

```
controller-1(config)# recorder-node device rn-alias
controller-1(config-recorder-node)# max-packet-age 30
```

This sets the maximum time to keep recorded packets to ***30*** minutes.



Note: Combine **Max Packet Age** with the packet removal policy to control when packets are deleted based on age rather than disk utilization alone.

9. (Optional) **Max Disk Utilization**: This defines the maximum disk utilization as a percentage between 5% and 95%. The **Disk Full Policy** (**`rolling-fifo`** or **`stop-and-wait`**) is enforced when reaching this value. If unset, the default maximum disk utilization is ***95%***; however, configure it, as required, using the following commands:

```
controller-1(config)# recorder-node device rn-alias
controller-1(config-recorder-node)# max-disk-utilization 80
```

10. (Optional) Disable unused or unneeded indexing configuration fields in subsequent recorder node queries. DMF enables all indexing fields by default. To disable a specific indexing option, enter the following commands from the ***config-recorder-node-indexing*** submode. To re-enable a disabled option, enter the command without the **no** prefix.

    Use the following command to enter the RN indexing submode:

    ```
    controller-1(config-recorder-node)# indexing
    controller-1(config-recorder-node-indexing)#
    ```

    Use the following commands to disable any unused fields in subsequent queries:

    - Disable MAC Source indexing: **no mac-src**
    - Disable MAC Destination indexing: **no mac-dst**
    - Disable outer VLAN ID indexing: **no vlan-1**
    - Disable inner/middle VLAN ID indexing: **no vlan-2**
    - Disable innermost VLAN ID indexing: **no vlan-3**
    - Disable IPv4 Source indexing: **no ipv4-src**
    - Disable IPv4 Destination indexing: **no ipv4-dst**
    - Disable IPv6 Source indexing: **no ipv6-src**
    - Disable IPv6 Destination indexing: **no ipv6-dst**
    - Disable IP Protocol indexing: **no ip-proto**
    - Disable Port Source indexing: **no port-src**
    - Disable Port Destination indexing: **no port-dst**
    - Disable MPLS indexing: **no mpls**
    - Disable Community ID indexing: **no community-id**
    - Disable MetaWatch Device ID: **no mw-device-id**
    - Disable MetaWatch Port ID: **no mw-port-id**

    For example, the following command disables indexing for the destination MAC address:

    ```
    controller-1(config-recorder-node-indexing)# no mac-src
    ```

11. Identify the RN interface by name in an out-of-band policy.

```
controller-1(config)# policy RecorderNodePolicy
controller-1(config-policy)# use-recorder-fabric-interface intf-1
controller-1(config-policy)#
```

12. Configure the DANZ Monitoring Fabric (DMF) policy to identify the traffic to send to the RN.

```
controller-1(config-policy)# 1 match any
controller-1(config-policy)# # filter-interface FilterInterface1
controller-1(config-policy)# # action forward
```

This example forwards all traffic received in the monitoring fabric on filter interface **FilterInterface1** to the RN interface. The following is the running-config for this example configuration:

```
recorder-fabric interface intf-1
description 'Delivery point for recorder-node'
recorder-interface switch 00:00:70:72:cf:c7:cd:7d ethernet37
policy RecorderNodePolicy
action forward
filter-interface FilterInterface1
use-recorder-fabric intf-1
1 match any
```

6.7.1.2 Authentication Token Configuration

Static authentication tokens are pushed to each Recorder Node (RN) as an alternative form of authentication in headless mode when the DANZ Monitoring Fabric (DMF) Controller is unreachable or by third-party applications that do not have or do not need DMF controller credentials to query the RN.

To configure the RN with a static authentication token, use the following commands:

```
controller-1(config)# recorder-node auth token mytoken
Auth : mytoken
Token : some_secret_string <--- secret plaintext token displayed once here
controller-1 (config)# show running-config recorder-node auth token
! recorder-node
recorder-node auth token mytoken $2a$12$cwt4PvsPySXrmMLYA.Mnyus9DpQ/bydGWD4LEhNL6xhPpkKNLzqWS <---hashed token shows in running
 config
```

The DMF Controller uses its hidden authentication token to query the RN. To regenerate the Controller authentication token, use the following command:

```
controller-1(config)# recorder-node auth generate-controller-token
```

6.7.1.3 Configuring the Pre-buffer

To enable the pre-buffer or change the time allocated, enter the following commands:

```
controller-1(config)# recorder-node device name
controller-1(config-recorder-node)# pre-buffer minutes
```

Replace ***name*** with the recorder node name. Replace ***minutes*** with the number of minutes to allocate to the pre-buffer.

6.7.1.4 Triggering a Recorder Node Event

To trigger an event for a specific Recorder Node (RN), enter the following command from enable mode:

```
controller-1# trigger recorder-node name event event-name
```

Replace ***name*** with the RN name and replace ***event-name*** with the name to assign to the current event.

6.7.1.5 Terminating a Recorder Node Event

To terminate a Recorder Node (RN) event, use the following command:

```
controller-1# terminate recorder-node name event event-name
```

Replace ***name*** with the RN name and replace ***event-name*** with the RN event name to terminate.

6.7.1.6 Viewing Recorder Node Events

To view recorder node events, enter the following command from enable mode:

```
controller-1# show recorder-node events
# Packet Recorder Time                          Event
-|---------------|-----------------------------|-------------------------------------------------------------------|
1 pkt-rec-740     2018-02-06 16:21:37.289000 UTC Pre-buffer event my-event1 complete. Duration 3 minute(s)
2 pkt-rec-740     2018-02-06 20:23:59.758000 UTC Pre-buffer event event2 complete. Duration 73 minute(s)
3 pkt-rec-740     2018-02-07 22:39:15.036000 UTC Pre-buffer event event-02-7/event3 complete. Duration 183 minute(s)
4 pkt-rec-740     2018-02-07 22:40:15.856000 UTC Pre-buffer event event5 triggered
5 pkt-rec-740     2018-02-07 22:40:16.125000 UTC Pre-buffer event event4/event-02-7 complete. Duration 1 minute(s)
6 pkt-rec-740     2018-02-22 06:53:10.216000 UTC Pre-buffer event triggered
```

6.7.2 Run Recorder Node Queries

> **Note:** The DANZ Monitoring Fabric (DMF) Controller prompt is displayed immediately after entering a query or replay request, but the query continues in the background. Attempting to enter another replay or query command before the previous command is completed, an error message is displayed.

6.7.2.1 Packet Replay

Enter the **`replay recorder-node`** command from enable mode to replay the packets recorded by a Recorder Node (RN).

```
controller-1# replay recorder-node name to-delivery interface filter stenographer-query
[realtime | replay-rate bps ]
```

The following are the options available with this command.

- ***name***: Specify the RN from which to replay the recorded packets.
- ***interface***: The DMF delivery interface name receiving the packets.
- ***stenographer-query***: The filter used to look up desired packets.
- (Optional) **real-time**: Replay the packets at the original rate recorded by the specified RN. The absence of this parameter will result in a replay up to the line rate of the RN interface.
- (Optional) **replay-rate *bps***: Specify the number of bits per second used for replaying the packets recorded by the specified RN. The absence of this parameter will result in a replay up to the line rate of the RN interface.

The following command shows an example of a replay command using the **to-delivery** option.

```
controller-1# replay recorder-node packet-rec-740 to-delivery eth26-del filter 'after 1m ago'
controller-1#
Replay policy details:
controller-1# show policy-flow | grep replay
1 __replay_131809296636625 packet-as5710-2 (00:00:70:72:cf:c7:cd:7d) 0 0 6400 1
in-port 47 apply: name=__replay_131809296636625 output: max-length=65535, port=26
```

6.7.2.2 Packet Data Query

Use a packet query to search the packets recorded by a specific Recorder Node (RN). The operation uses a Stenographer query string to filter only the interesting traffic. The query returns a URL to download and analyze the packets using Wireshark or other packet-analysis tools.

From enable mode, enter the **`query recorder-node`** command.

```
switch # query recorder-node name packet-data filter stenographer-query
```

The following is the meaning of each parameter:

- ***name***: Identify the RN.
- **packet-data filter *stenographer-query***: Look up only the packets that match the specified Stenographer query.

The following example illustrates the results returned:

```
switch # query packet-recorder hq-hmf-packet-recorder-1 packet-data filter "after 1m ago and src host 8.8.
8.8"
------------------------------------------ Packet Query Results ------------------------------------------
Individual URL(s) : /pcap/__packet_recorder__/hq-hmf-packet-recorder-1-2018-10-19-08-19-59-40bbddde.
pcap
- Error(s) -
None.
Past packet queries can be referenced on the controller using the show packet-capture files policy
__packet_recorder__ command. Any HTTP client can be used to download the files using the URLs indicated.
switch# show packet-capture files policy __packet_recorder__ | head
------------------------------------------------------------------ Packet Capture Files for Selected Policy ------------------------------------------------------------------
#  File Name                                                 File Size Last Modified           URL
--|---------------------------------------------------------|---------|-------------------------|-----------------------------------------------------------------------
```

6.7.2.3 Packet Object Query

The packet object query extracts unencrypted HTTP objects from packets matching the given stenographer filter. To run a packet object query, run the following query command:

```
switch# query recorder-node bmf-integrations-pr-1 packet-object filter 'after 5m ago'
```

The following example illustrates the results returned:

```
switch# query recorder-node bmf-integrations-pr-1 packet-object filter 'after 1m ago'
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Packet Object Query Results ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Coalesced URL     : /pcap/__packet_recorder__/coalesced-bmf-2022-11-21-14-27-56-67a73ea9.tgz
Individual URL(s) : /pcap/__packet_recorder__/bmf-integrations-pr-1-2022-11-21-14-27-55-598f5ae7.tgz
```

Untar the folder to extract the HTTP objects.

6.7.2.4 Size Query

Use a size query to analyze the number of packets and the total size recorded by a specific Recorder Node (RN). The operation uses a Stenographer query string to filter only the interesting traffic.

Enter the **`query recorder-node`** command from enable mode to run a size query.

```
# query recorder-node name size filter stenographer_query
```

The following is the meaning of each parameter:

- ***name***: Identify the RN.
- **size filter *stenographer-query***: Analyze only the packets that match the specified Stenographer query.

The following example illustrates the results returned:

```
switch# query recorder-node hq-bmf-packet-recorder-1 size filter "after 1m ago and src host
 8.8.8.8"
~ Summary Query Results ~
# Packets : 66
Size      : 7.64KB
~ Error(s) ~
None.
```

6.7.2.5 Window Query

Use a window query to analyze the oldest and most recent available packets recorded by a specific Recorder Node (RN).

Enter the **query recorder-node** command from enable mode to run a window query.

```
switch# query recorder-node name window
```

The following is the meaning of each parameter:

- ***name***: Identify the RN.

The following example illustrates the results returned:

```
switch# query recorder-node hq-bmf-packet-recorder-1 window
~~~~~~~~~~~~~~ Window Query Results ~~~~~~~~~~~~~~
Oldest Packet Available : 2020-07-30 05:01:08 PDT
Newest Packet Available : 2020-10-19 08:14:21 PDT
~ Error(s) ~
None.
```

6.7.2.6 Stopping a Query

Use the abort **recorder-node** command to stop the query running on the specified Recorder Node (RN). From enable mode, enter the following command:

```
controller-1# abort recorder-node name filter string
```

Replace ***name*** with the RN name, and use the **filter** keyword to identify the specific filter used to submit the query. If the specific running query is unknown, use an empty-string filter of **""** to terminate any running query.

```
controller-1# abort recorder-node hq-bmf-packet-recorder-1 filter ""
Abort any request with the specified filter? This cannot be undone. enter "yes" (or "y") to
 continue:
yes
Result : Success
~ Error(s) ~
None.
```

6.7.3 Viewing Query History

View Recorder Node (RN) submitted queries using the CLI.

To display query history, enter the following command:

```
dmf-controller> show recorder-node query-history
#   Packet Recorder Query          Type                     Start
 Duration
---|---------------|--------------|------------------------|------------------------------|----
----|
1   HW-PR-2         after 10m ago  analysis-hosts           2019-03-20 09:52:38.021000 PDT 3428
2   HW-PR-1         after 10m ago  analysis-hosts           2019-03-20 09:52:38.021000 PDT 3428
3   HW-PR-2         after 10m ago  abort                    2019-03-20 09:52:40.439000 PDT 711
4   HW-PR-1         after 10m ago  abort                    2019-03-20 09:52:40.439000 PDT 711
---------------------------------output truncated-------------------------------------------
--------
```

6.7.4 Using RBAC to Manage Access to the DMF Recorder Node

Use Role-Based Access Control (RBAC) to manage access to the DANZ Monitoring Fabric (DMF) Recorder Node (RN) by associating the RN with an RBAC group.

To restrict access for a specific RN to a specific RBAC group, use the CLI as described in the following instructions.

6.7.4.1 RBAC Configuration Using the CLI

1. Identify the group to associate the Recorder Node (RN).
   Enter the following command from config mode on the active DANZ Monitoring Fabric (DMF) controller:

```
controller-1(config)# group test
controller-1(config-group)#
```

2. Associate one or more RNs with the group.

   Enter the following CLI command from the ***config-group*** submode:

```
controller-1(config-group)# associate recorder-node device-name
```

   Replace ***device-name*** name with the RN name, as in the following example:

```
controller-1(config-group)# associate recorder-node HW-PR-1
```

6.7.5 View Information About a Recorder Node

This section describes monitoring and troubleshooting the Recorder Node (RN) status and operation. The RN stores packets on the main hard disk and the indices on the SSD volumes.

6.7.5.1 Viewing the Recorder Node Interface

To view information about the RN interface information, use the following command:

```
controller-1(config)# show topology recorder-node
# DMF IF       Switch IF  Name       State Speed  Rate Limit
-|------------|----------|----------|-----|------|----------|
1 RecNode-Intf Arista7050 ethernet1  up    25Gbps -
```

6.7.5.2 Viewing Recorder Node Operation

```
controller-1# show recorder-node device packet-rec-740 interfaces stats
Packet Recorder Name Rx Pkts       Rx BytesRx Drop  Rx Errors Tx Pkts  Tx Bytes   Tx Drop Tx Errors
---------------|----|-------------|---------------|--------|---------|--------|----------|-------|---------|
packet-rec-740  pri1 2640908588614 172081747460802 84204084 0 24630503 3053932660 0       0
```

Information about a Recorder Node (RN) interface used as a delivery port in a DANZ Monitoring Fabric (DMF) out-of-band policy appears in a list. It lists RN interfaces as dynamically added delivery interfaces.

```
Ctrl-2(config)# show policy PR-policy
Policy Name                          : PR-policy
Config Status                        : active - forward
Runtime Status                       : installed
Detailed Status                      : installed - installed to forward
Priority                             : 100
Overlap Priority                     : 0
# of switches with filter interfaces   : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces  : 0
# of filter interfaces                : 1
# of delivery interfaces              : 1
# of core interfaces                  : 0
# of services                         : 0
# of pre service interfaces           : 0
# of post service interfaces          : 0
Push VLAN                            : 1
Post Match Filter Traffic            : 1.51Gbps
Total Delivery Rate                  : 1.51Gbps
Total Pre Service Rate               : -
Total Post Service Rate              : -
Overlapping Policies                 : none
Component Policies                   : none
Installed Time                       : 2023-09-22 12:16:55 UTC
Installed Duration                   : 3 days, 4 hours
~ Match Rules ~
# Rule
-|----------|
1 1 match any

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF      Switch              IF Name   State Dir Packets     Bytes          Pkt Rate Bit Rate Counter Reset Time
-|----------|-------------------|---------|-----|---|-----------|--------------|--------|--------|------------------------------|
1 Lab-traffic Arista-7050SX3-T3X5 ethernet7 up    rx  97831460642 51981008309480 382563   1.51Gbps 2023-09-22 12:16:55.738000 UTC

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF          Switch              IF Name    State Dir Packets     Bytes          Pkt Rate Bit Rate Counter Reset Time
-|---------------|-------------------|----------|-----|---|-----------|--------------|--------|--------|------------------------------|
1 PR-intf Arista-7050SX3-T3X5 ethernet35 up    tx  97831460642 51981008309480 382563   1.51Gbps 2023-09-22 12:16:55.738000 UTC

~ Service Interface(s) ~
None.

~ Core Interface(s) ~
None.

~ Failed Path(s) ~
None.
Ctrl-2(config)#
```

6.7.5.3 Viewing Errors and Warnings

The following table lists the errors and warnings a recorder node may display. In the CLI, display these errors and warnings by entering the following commands:

- **`show fabric errors`**
- **`show fabric warnings`**
- **`show recorder-node errors`**
- **`show recorder-node warnings`**

Table 4: Errors and Warnings

Type	Condition	Cause	Resolution
Error	Recorder Node (RN) management link down.	RN has not received controller LLDP.	Wait 30s if the recorder node is newly configured. Verify it is not connected to a switch port that is a DANZ Monitoring Fabric (DMF) interface.
Error	RN fabric link down.	Controller has not received RN LLDP.	Wait 30s if recorder node is newly configured. Check it is online otherwise.
Warning	Disk/RAID health degraded.	Possible hardware degradation.	Investigate specific warning reported. Could be temperature issue. Possibly replace indicated disk soon.
Warning	Low disk space.	Packet or index disk space has risen above threshold.	Prepare for disk full soon.
Warning	Disk full.	Packet or index disk space is full. Packets are being dropped or rotated depending on removal policy.	Do nothing if removal policy is rolling-FIFO. Consider erasing packets to free up space otherwise.
Warning	Recorder misconfiguration on a DMF interface.	A recorder node has been detected in the fabric on a switch interface that is configured as a filter or delivery interface.	Remove the conflicting interface configuration, or re-cable the recorder node to a switch interface not defined as a filter or delivery interface.

6.7.6 Changing the Recorder Node Default Configuration

Configuration settings are automatically downloaded to the Recorder Node (RN) from the DANZ Monitoring Fabric (DMF) Controller, eliminating the need for box-by-box configuration. However, the option exists to override the default configuration for the RN from the ***config-recorder-node*** submode for any RN.



Note: These options are available only from the CLI, not the DMF Controller GUI.

To change the CLI mode to ***config-recorder-node***, enter the following command from config mode on the active DMF controller:

```
controller-1(config)# recorder-node device instance
```

Replace ***instance*** with the alias to use for the RN. This alias is affiliated with the MAC hardware address using the **mac** command.

Use any of the following commands from the **config-recorder-node** submode to override the default configuration for the associated RN:

- **banner**: Set the RN pre-login banner message
- **mac**: Configure the MAC address for the RN

Additionally, the option exists to override the configurations shown below to use values specific to the RN or used in a merge-mode along with the configuration inherited from the DMF controller:

- **ntp**: Configure RN to override default timezone and NTP parameters.
- **snmp-server**: Configure RN SNMP parameters and traps.
- **logging**: Enable RN logging to Controller.
- **tacacs**: Set TACACS defaults, server IP address(es), timeouts and keys.

Use the following commands from the ***config-recorder-node*** submode to change the default configuration on the RN:

- **ntp override-global**: Override global time configuration with RN time configuration.
- **snmp-server override-global**: Override global SNMP configuration with RN SNMP configuration.
- **snmp-server trap override-global**: Override global SNMP trap configuration with RN SNMP trap configuration.
- **logging override-global**: Override global logging configuration with packet recorder logging configuration.
- **tacacs override-global**: Override global TACACS configuration with RN TACACS configuration.

To configure the RN to work in a merge mode by merging its specific configuration with that of the DMF Controller, execute the following commands in the ***config-recorder-node*** submode:

- **`ntp merge-global`**: Merge global time configuration with RN time configuration.
- **`snmp-server merge-global`**: Merge global SNMP configuration with RN SNMP configuration.
- **`snmp-server trap merge-global`**: Merge global SNMP trap configuration with RN SNMP trap configuration.
- **`logging merge-global`**: Merge global logging configuration with RN logging configuration.

TACACS configuration does not have a merge option. It can either be inherited from the DMF Controller or overridden to use only the RN-specific configuration.

6.7.7 Large PCAP Queries

Access the RN via a web browser to run large PCAP queries to the Recorder Node (RN). This allows running packet queries directly to the RN without specifying the maximum byte or packet limit for the PCAP file (which is required when executing the query from the DANZ Monitoring Fabric (DMF) Controller).

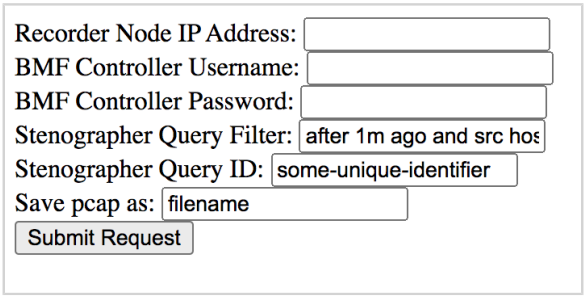
To access the RN directly, use the URL **`https://`*`RecorderNodeIP`*** in a web browser, as shown below:

Figure 6-90: URL to Recorder Node



The following page will be displayed:

Figure 6-91: Recorder Node Page



- **Recorder Node IP Address**: Enter the target RN IP address.
- **DMF Controller Username**: Provide the DMF Controller username.
- **DMF Controller Password**: Provide the password for authentication.
- **Stenographer Query Filter**: Use the query filter to filter the query results to look for specific packets. For example, to search for packets with a source IP address of ***10.0.0.145*** in the last 10 minutes, use the following filter:

  ```
  after 10m ago and src host 10.0.0.145
  ```

- **Stenographer Query ID**: Starting in DMF 8.0, a Universally Unique Identifier (UUID) is required to run queries. To generate a UUID, run the following command on any Linux machine and use the result as the Stenographer query ID:

  ```
  $ uuidgen
  b01308db-65f2-4d7c-b884-bb908d111400
  ```

- **Save pcap as**: Provide the file name for this PCAP query result.
- **Submit Request**: Sends a query to the specified RN and saves the PCAP file with the provided file name to the default download location for the browser.

6.7.8 Recorder Node Management Migration L3ZTN

After completing the first boot (initial configuration), remove the Recorder Node (RN) from the old Controller and point it to a new Controller via the CLI in the case of a Layer-3 topology mode.

> **Note:** For appliances to connect to the DANZ Monitoring Fabric (DMF) Controller in Layer-3 Zero Touch Network (L3ZTN) mode, configure the DMF Controller deployment mode as **`pre-configure`**.

To migrate management to a new Controller, follow the steps below:

1. Remove the RN and switch from the old Controller using the commands below:

```
controller-1(config)# no recorder-node device RecNode
```

```
controller-1(config)# no switch Arista7050
```

2. Add the switch to the new Controller.
3. SSH to the RN and configure the new Controller IP using the **`zerotouch l3ztn controller-ip`** command:

```
controller-1(config)# zerotouch l3ztn controller-ip 10.2.0.151
```

4. After pointing the RN to use the new Controller, reboot the RN.
5. Once the RN is back online, the DMF Controller receives the ZTN request.

```
controller-1(config)# show zerotouch request
# Request-history             Ip address Action              Timestamp                  Result          Message
-|--------------------------|---------|--------------------|--------------------------|---------------|-----------------------------
```

6.7.9 Recorder Node Show Commands

The following commands are available from the Recorder Node (RN):

Use the **`show version`** command to view the version and image information that RN is running on.

```
RecNode(config)# show version
Controller Version : DMF Recorder Node 8.1.0 (bigswitch/enable/dmf-8.1.x #5)
RecNode(config)#
```

Use the **`show controllers`** command to view the connected DANZ Monitoring Fabric (DMF) controllers to the recorder node.



Note: All cluster nodes appear in the command output if the RN is connected to a DMF Controller cluster.

```
RecNode(config)# show controllers
controller            Role   State     Aux
----------------------|------|---------|---|
tcp://10.106.8.2:6653 master connected 0
tcp://10.106.8.3:6653 slave  connected 0
tcp://10.106.8.3:6653 slave  connected 1
tcp://10.106.8.3:6653 slave  connected 2
tcp://10.106.8.2:6653 master connected 1
tcp://10.106.8.2:6653 master connected 2
RecNode(config)#
```

6.7.10 Ability to Deduplicate Packets - Query from Recorder Node

For Recorder Node queries, the recorded packets matching a specified query filter may contain duplicates when packet recording occurs at several different TAPs within the same network; i.e., as a packet moves through the network, it may be recorded multiple times. The dedup feature removes duplicate packets from the query results. By eliminating redundant information, packet deduplication improves query results' clarity, accuracy, and conciseness. Additionally, the dedup feature significantly reduces the size of query results obtained from packet query types.

6.7.10.1 Deduplicate Packets

In the DANZ Monitoring Fabric (DMF) Controller CLI, packet deduplication is available for the packet data, packet object, size, and replay query types. Deduplication is **off** by default for these queries. Add the **dedup** option to the end of the query command after all optional values (if any) have been selected to enable deduplication.

The following are command examples of enabling deduplication.

Enabling deduplication for a size query:

```
controller# query recorder-node rn size filter "before 5s ago" dedup
```

Enabling deduplication for a packet data query specifying a limit for the size of the PCAP file returned in bytes:

```
controller# query recorder-node rn packet-data filter "before 5s ago" limit-bytes 2000 dedup
```

Enabling deduplication for a replay query:

```
controller# replay recorder-node rn to-delivery dintf filter "before 5s ago" dedup
```

Enabling deduplication for a replay query specifying the replay rate:

```
controller# replay recorder-node rn to-delivery dintf filter "before 5s ago" replay-rate 100 dedup
```

Specify a time window (in milliseconds) for deduplication. The time window defines the time required between timestamps of identical packets to no longer be considered duplicates of each other. For example, for a time window of 200 ms, two identical packets with timestamps that are 200 ms (or less) apart are duplicates of each other. In contrast, if the two identical packets had timestamps more than 200 ms apart, they would not be duplicates of each other.

The time window must be an integer between 0 and 999 (inclusive) with a default time window of 200 ms when deduplication is enabled and no set time window value.

To configure a time window value, use the **dedup-window** option followed by an integer value for the time window after the **dedup** option.

```
controller# query recorder-node rn size filter "before 5s ago" dedup dedup-window 150
```

6.7.11 Enable Egress sFlow

Enable egress sFlow®* to sample traffic sent to any DANZ Monitoring Fabric (DMF) Recorder Node (RN) attached to the fabric. Examining these sampled packets on a configured sFlow collector allows the identification of post-match-rule flows recorded by the RNs without performing a query against the RNs. While not explicitly required, Arista Networks highly recommends using the DMF Analytics Node (AN) as the configured sFlow collector, as it can automatically identify packets sampled utilizing this feature.

Platform Compatibility

All platforms apart from the following series:

- DCS-7280R
- DCS-7280R2
- DCS-7500R
- DCS-7020
- DCS-7050X4

Configuration

The egress sFlow feature requires a configured sFlow collector. After configuring the sFlow collector, enter the following command from the config mode to enable the feature:

```
Controller-1(config)# recorder-node sflow
```

To disable the feature, enter the command:

```
Controller-1(config)# no recorder-node sflow
```

6.8 Considerations and Limitations

* sFlow® is a registered trademark of Inmon Corp.

6.8.1 Deduplication Limitations

Expect a query with packet deduplication enabled to take longer to complete than packet deduplication disabled. Hence, packet deduplication, by default, is off.

The maximum time window value permitted is 999 ms to ensure that TCP retransmissions are not regarded as duplicates, assuming that the receive timeout value for TCP retransmissions (of any kind) is at least 1 second. If the receive timeout value is less than 1 second (particularly, exactly 999 ms or less), then it is possible for TCP retransmissions to be regarded as duplicates when the time window value used is larger than the receive timeout value.

Due to memory constraints, removing some duplicates may not occur as expected. This scenario is likely to occur if a substantial amount of packets match the query filter, which all have timestamps within the specified time window from each other. We refer to this scenario as the query having exceeded the packet window capacity. To mitigate this from occurring, decrease the time window value or use a more specific query filter to reduce the number of packets matching the query filter at a given time.

6.8.2 Guidelines and Limitations for Enabling Egress sFlow

Consider the following guidelines and limitations while enabling Egress sFlow:

- The Egress sFlow support for the Recorder Nodes (RN) feature requires a configured sFlow collector in a fabric configured to allow sFlows.
- If a packet enters a switch through a filter interface with sFlow enabled and exits through a port connected to an RN while the feature is enabled, only one sFlow packet (i.e., the ingress sFlow packet) is sent to the collector.
- The Egress sFlow feature does not identify which RN has recorded a given packet in a fabric when there are multiple RNs. This is fine in a normal case as the queries are issued to the RNs in aggregate rather than to individual RNs, and hence, the information that any RN has received a packet is sufficient. In some cases, it may be possible to make that determination from the outport of the sFlow packet, but that information may not be available in all cases. This is an inherent limitation of egress sFlow.
- An **enabled** egress sFlow feature captures the packets sent to any RN with recording enabled, regardless of whether the RN is actively recording or not.

6.8.3 Recorder Node Recording State API Limitations

The **`ready`** state only occurs after the recording application has finished initializing if no recordable traffic has been received yet. The recording application must undergo its initialization process whenever the RN is rebooted, restarted, or after restarting the RN application from the DMF Controller. If the RN is in the **`active`** state and stops receiving packets, it will not regress into the **`ready`** state; it will remain in the **`active`** state.

6.9 Recorder Node Support for CVaaS

The Recorder Node (RN) supports being managed by CloudVision (CV) on-prem starting DMF 8.7.0. This feature extends support to CVaaS starting DMF 8.8.0. Recorder Node was not supported with CVaaS before 8.8.0 because of an RN requirement to store the query results file in CV while archiving the query results. However, this was not permitted on CVaaS as these files might contain data that cannot be stored in a cloud service. This feature supports CVaaS by allowing the RN to store query result files.



Note: This feature is available in all RN management modes (DMF, CV on-prem, and CVaaS); however, only CVaaS is currently using it.

Query Files and Storage

RN stores query files in the query archive directory. The system mounts the directory (`/var/lib/query-archive`) to a new storage volume called the archive volume. This volume (`/dev/flvg/query-archive`) is part of the RN's main disk and can be 10 or 50GB in size, depending on the size of the main disk. This directory is mapped to a new file server on the RN, allowing retrieval of query files from the RN via a URL.

The RN manages stored query files without user input. The system automatically deletes files from the archive older than 7 days, deleting any files when the archive volume does not have enough space for a new query file to be stored. In this case, it first deletes the oldest files until enough space becomes available. Metadata queries require at least 256MB to be available on the volume, whereas the user can set this value for packet queries.

Stored Files State API

The RN state API includes a list of all query files currently stored in the archive. This new list is available at `/applications/recorder-node/state/query/stored-files`. Each entry of the list includes the following information describing the query file:

- `file-name`: Name of stored query result file
- `query-id`: ID of associated query request
- `creation-time`: ISO8601 timestamp of when the file was created
- `file-size`: Size of file in bytes



Note: When a file is deleted by the RN, the corresponding entry is removed from this list.

Query RPC API

All RN query RPC APIs include a new input parameter called `storage-method` for selecting the archiving method to use for the request. The new parameter is an enumeration that has the following choices:

- no-store: (default) Result will not be archived
- local: Store query result file on the local archive volume
- upload: Upload query result file to a remote file server

For packet queries, an additional input parameter called `max-result-size` is available. This parameter controls the maximum size (in bytes) of the query result, which by default is 2GB.



Note: The max result size is applied for packet queries regardless of the storage method selected.

The RPC output for all queries now includes a `file-url` entry which is the location where the query result file is stored. This is an empty string when no storage method is selected. For the upload method, this is the upload URL provided in the query input. For the local method, this is a relative URL for downloading the file from the file server on the RN.

Platform Compatibility

This feature is available for both physical and virtual Recorder Nodes. The only difference between the two deployments is the size of the archive volume. For a physical RN, the volume size is 50GB. For the virtual RN, the volume size might be limited to 10GB in scenarios where the main disk (the disk of the platform where the virtual RN is deployed) is not sufficiently large.

Show Commands

The stored query files can be viewed from the RN CLI using show query stored-files. Below is an example of the output from this command.

```
pr1> show query stored-files
# Creation time           File name                                                    File
 size Query id
-|----------------------|-------------------------------------------------------------|
---------|--------------------------------|
1 2025-06-12 17:56:12 UTC size_2025-06-12T17-56-12Z_luwIwp4rnInbkb1j24BW8pj3ua_FwaYV.json 42B
  luwIwp4rnInbkb1j24BW8pj3ua_FwaYV
2 2025-06-12 18:02:14 UTC size_2025-06-12T18-02-14Z_9AUJ277ZmpCTr07NszoEC0DCFggzixU2.json 43B
  9AUJ277ZmpCTr07NszoEC0DCFggzixU2
```

Troubleshooting

- For issues with accessing the file server, view the access log on the RN at: `/var/log/nginx/packet-recorder.access.log`.
- For archiving errors returned by the query RPC, view the RN floodlight log at: `/var/log/floodlight/floodlight.log`. Error, warning, and info logs relating to this feature in the floodlight log have log ID prefix `RNQRY`. The floodlight log can be filtered by this prefix to find logs relating to this feature for troubleshooting purposes.
- For issues with the setup of the archive volume or directory, view the `storage.service` log using the following command in RN bash: **`sudo journalctl -u storage`**.

RPC Errors

The following are new RPC errors that can be returned by a query if the local storage method is selected:

- `400: Invalid max result size, value exceeds total size of storage volume`
  - If this occurs for a packet query where the max-result-size parameter has been configured, please try decreasing the value of this parameter. The value of the parameter should not exceed the size of the volume. The size of the volume is 10GB or 50GB based on the platform used (VM or physical RN).
  - If this occurs for a query where either the max-result-size parameter has not been configured or is not configurable, please try rebooting the RN.
- `500: Query file archive is unavailable, see floodlight log for details`
  - This error is due to the file archive being unhealthy in some capacity. The error log associated with this error has log ID `RNQRY7006`. The exception included in this log indicates the specific issue resulting in the archive being considered unhealthy. Please try rebooting the RN.



Note: For issues where rebooting the device is suggested, the assumption is that a user has made changes to the archive in some way (e.g., reduced the volume size, removed the volume, moved the archive directory). If this is not the case, please create a support bundle and contact Arista TAC.

Considerations

- If a file that the RN is trying to delete is currently open, whether a user manually opened it or the system opened it because it is currently being downloaded, the delete operation will fail. In this case, the RN does not retry deleting the file, but the system removes it from the stored files list. So, it will appear as if the file has been deleted, but in fact, it has not been deleted. Restarting the RN will re-initialize the stored files list based on the contents of the archive directory, at which point the removed file is tracked again in the list.
- Users can manually delete files from the archive directory. However, this is not recommended unless necessary, as it causes the stored files list to be out of sync with the filesystem for a short time. Manually deleted files are removed from the stored files list within an hour of being deleted.
- Users can store other files within the archive directory if desired. However, this is not recommended unless necessary since the RN only manages query files within this directory. The RN ignores all other files within this directory, so they will never be deleted or modified in any way. This is problematic if these files occupy a large amount of space on the volume, which may lead to queries failing due to a lack of available space.

Chapter 7

Link Aggregation

This chapter describes configuring link aggregation groups between switches, switches, and tools or between switches and taps.

7.1 Configuring Link Aggregation

Link aggregation combines multiple LAN links and cables in parallel. Link aggregation provides a high level of redundancy and higher transmission speed.



Note: When connecting a Link Aggregation Group (LAG) to a DMF Service Node appliance, member links can be connected to multiple DMF Service Node appliances with data ports of the same speed.

DMF provides a configurable method of hashing for load distribution among LAG members. The enhanced hashing algorithm automatically assigns the best hashing type for the switch and traffic. This setting allows the manual selection of the packet types and fields used for load distribution among the members of a port-channel interface. Enhanced mode and symmetric hashing are enabled by default for the supported switch platforms. With symmetric hashing, bidirectional traffic between two hosts going out on a port channel is distributed on the same member port.

The default hashing option uses the best available packet header field that applies to each packet, and that is supported by the switch. These fields can include the following:

- IPv4
- IPv6
- MPLS (disabled by default)
- L2GRE packet

If none of these headers apply, DMF uses Layer-2 header fields (source MAC address, destination MAC address, VLAN-ID, and ethertype) to distribute traffic among the LAG member interfaces. Hashing on the following packet header fields is enabled by default:

- **`hash l2 dst-mac eth-type src-mac vlan-id`**
- **`hash ipv4 dst-ip src-ip`**
- **`hash ipv6 dst-ip src-ip`**
- **`hash l2gre inner-l3 dst-ip src-ip`**
- **`hash symmetric`**



Note: DMF treats VN-tagged and QinQ packets as L2 packets and uses Layer-2 headers to distribute traffic among LAG member interfaces for these packets.

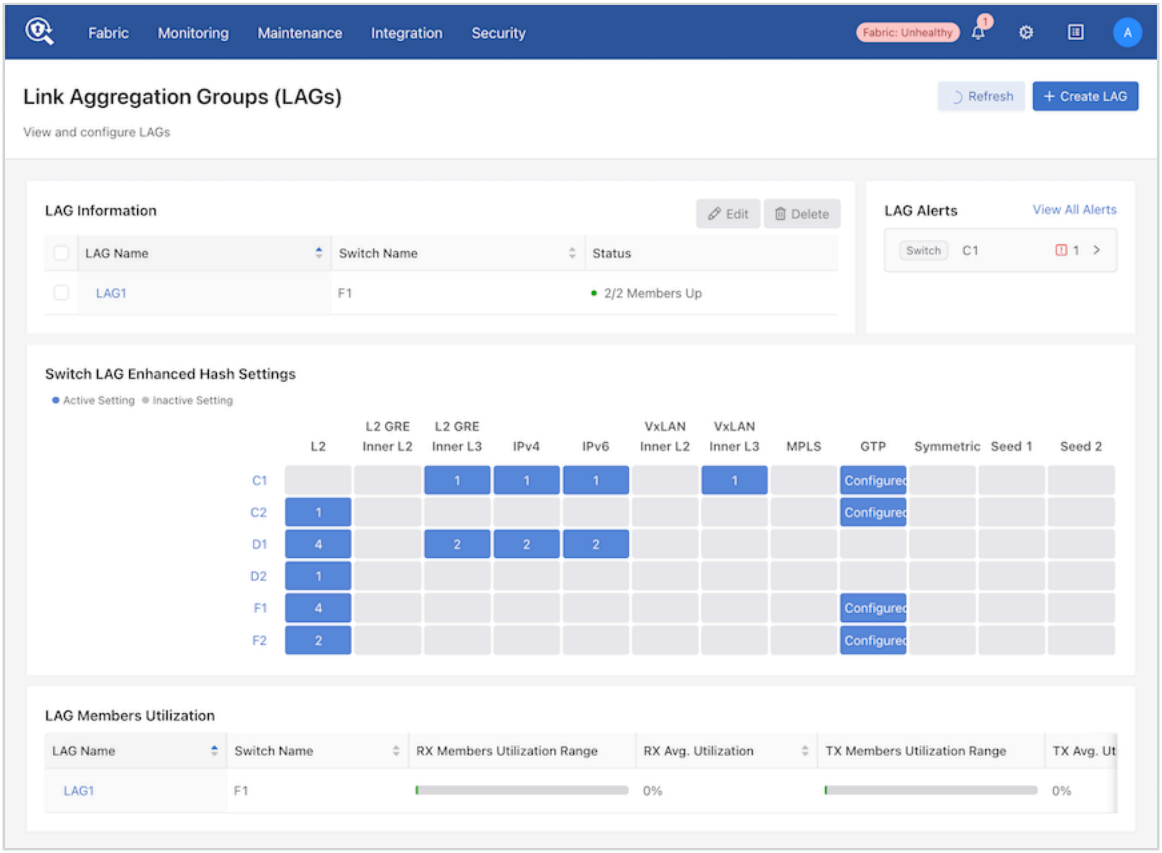
7.1.1 Configure Link Aggregation Groups

Overview

A new DMF LAGs Page introduces an improved workflow and additional functionality.

Navigate to **Fabric** > **LAGs**.

Figure 7-1: Lag Aggregation Groups Dashboard



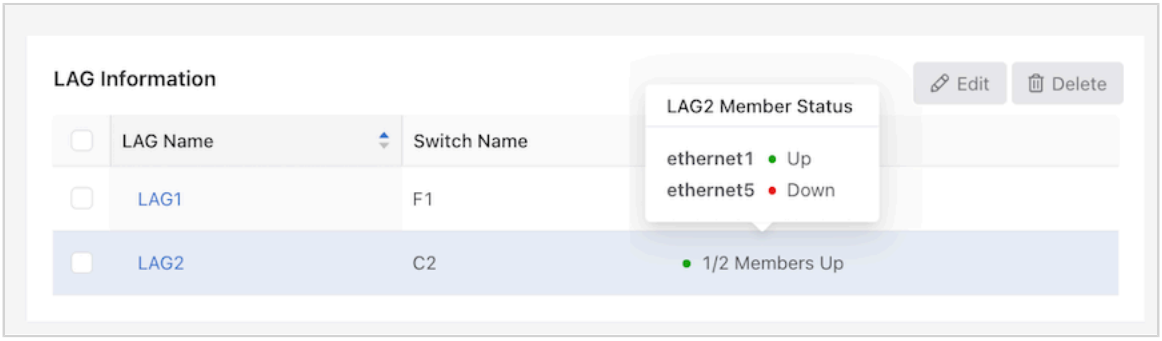
Main Page

The main page contains four components:

- **LAG Information**
- **LAG Alerts**
- **Switch LAG Enhanced Hash Settings**
- **LAG Members Utilization**

LAG Information

Figure 7-2: LAG Information



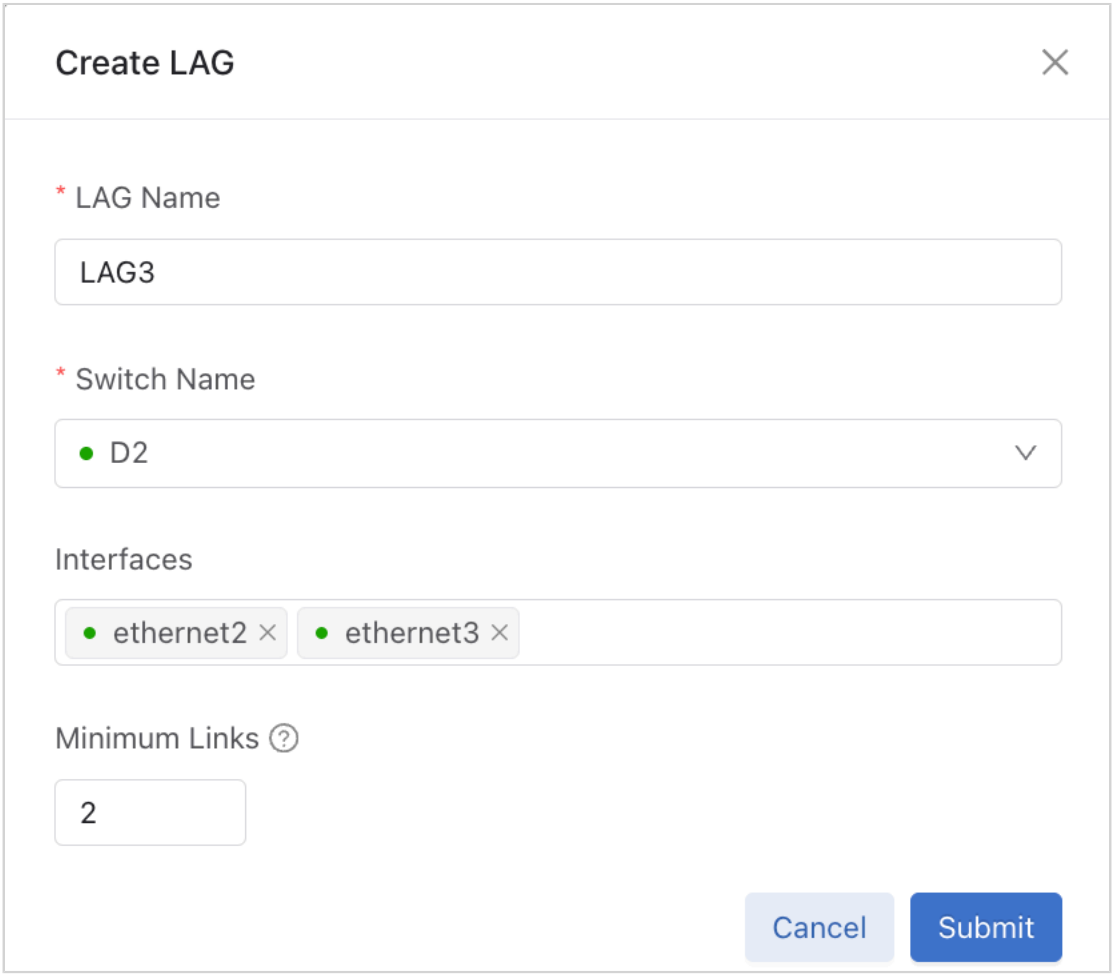
The LAG Information table displays all the LAGs configured in DMF and each member's status. A row contains the **LAG Name**, **Switch Name**, and **Member Status**.

Hover over a member under **Status** to view the status of each member interface.

Use the checkbox to **Edit** or **Delete** a LAG entry. Select the **LAG Name** link to open the properties tab of the LAG details panel.

Create or Edit LAG

Figure 7-3: Create LAG



Select **Create LAG** to open the configuration panel.

Enter a **LAG Name** and the **Switch Name** from the drop-down list, and choose the member **Interfaces** from that switch.

As an option, set the **Minimum Links** configuration required for the LAG to stay up. Refer to Minimum Link and Activation Configuration for Lag Interfaces for more information.

Select **Submit** to create the LAG.

Tip: If no configured LAGs exist, select **Create LAG** on the main page.

To edit a LAG, choose it in the Information table and select **Edit**.

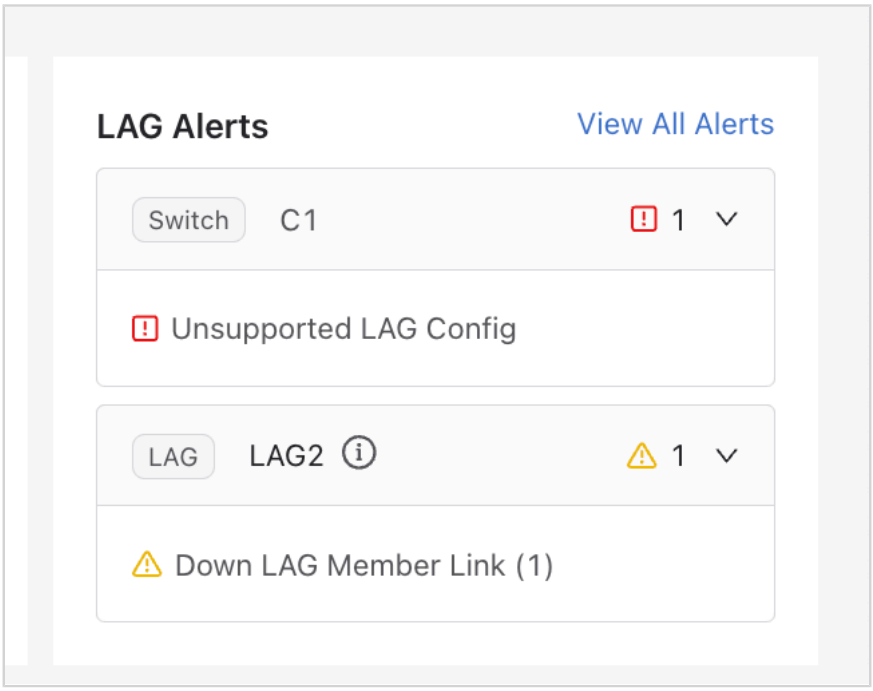
While the LAG Name and switch name are not editable, modifying member **Interfaces** and **Minimum Links** values is.

Only one LAG can be edited at a time; however, the system supports deleting multiple LAGs simultaneously.

Select **Submit** to commit the changes to the LAG.

LAG Alerts

Figure 7-4: LAG Alerts



LAG Alerts displays all LAG Alerts and LAG-related Switch Alerts. The **Switch** or **LAG** tag indicates the type of alert. **Expand** each item to view the alert description and the number of occurrences.

LAG Enhanced Hash Grid

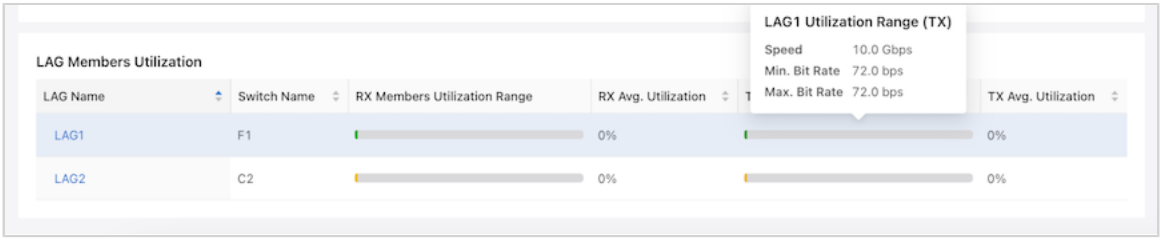
Figure 7-5: Grid



The **Switch Lag Enhanced Hash Settings** grid displays a switch's current LAG-enhanced hash settings across various hash fields and categories. If a cell is gray, there are no currently active sub-settings. Hover over the blue cell to view active fields. Select a **switch name** link to open the Enhanced Hash Configuration tab of the LAG Details pane.

LAG Members Utilization

Figure 7-6: Utilization



The **LAG Members Utilization** table summarizes the utilization for each LAG.

In the Utilization Range column, a colored bar indicates the start of the minimum utilization and the end of the maximum utilization of its members. Hover over the bar to view the Speed, minimum bit Rate, and maximum bit Rate of the LAG for each direction. The bar changes color based on LAG alerts:

- Green – indicates no alerts.
- Yellow – only warnings.
- Red – there is more than one alert.

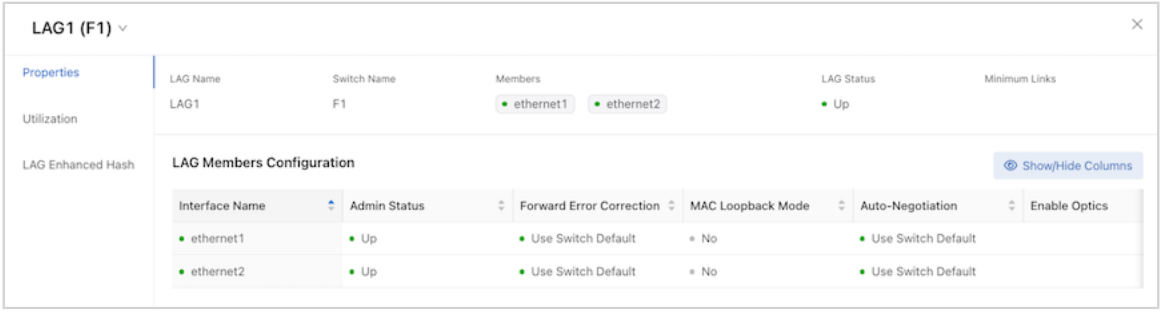
Select the **LAG Name** link to open the Utilization tab of the LAG Details pane.

Details Pane

Select a **LAG** or **switch link** to open the details pane. Choose a **LAG** to view its details. When choosing a switch link in the Enhanced Hash grid, the first LAG on that switch is selected. If there isn't a LAG on the switch, the system displays an empty string, and the Enhanced Hash opens for that switch. The Properties and Utilization tab will be disabled. Select a different LAG to access the other tabs.

Properties

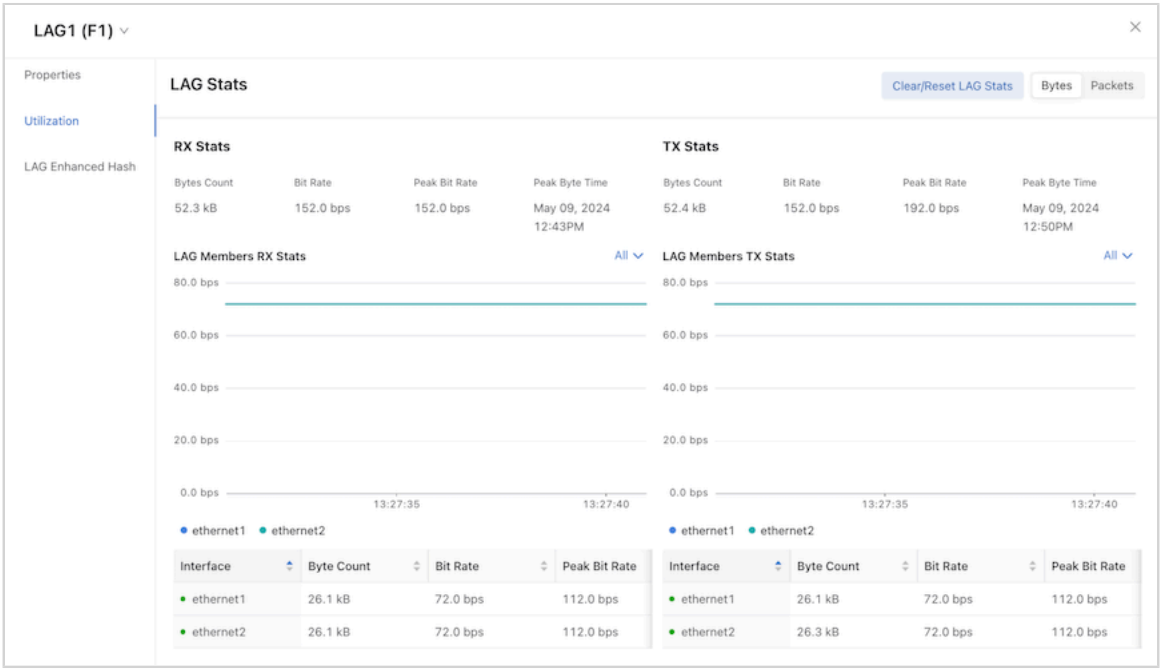
Figure 7-7: Properties



The LAG Name, Switch Name, Members tags, and Minimum Link values appear in the descriptions section. The Members Configuration table shows the configuration of each member interface. Use **Show/Hide Columns** to determine which columns to display.

Utilization

Figure 7-8: Utilization



Utilization displays the utilization statistics for the LAG.

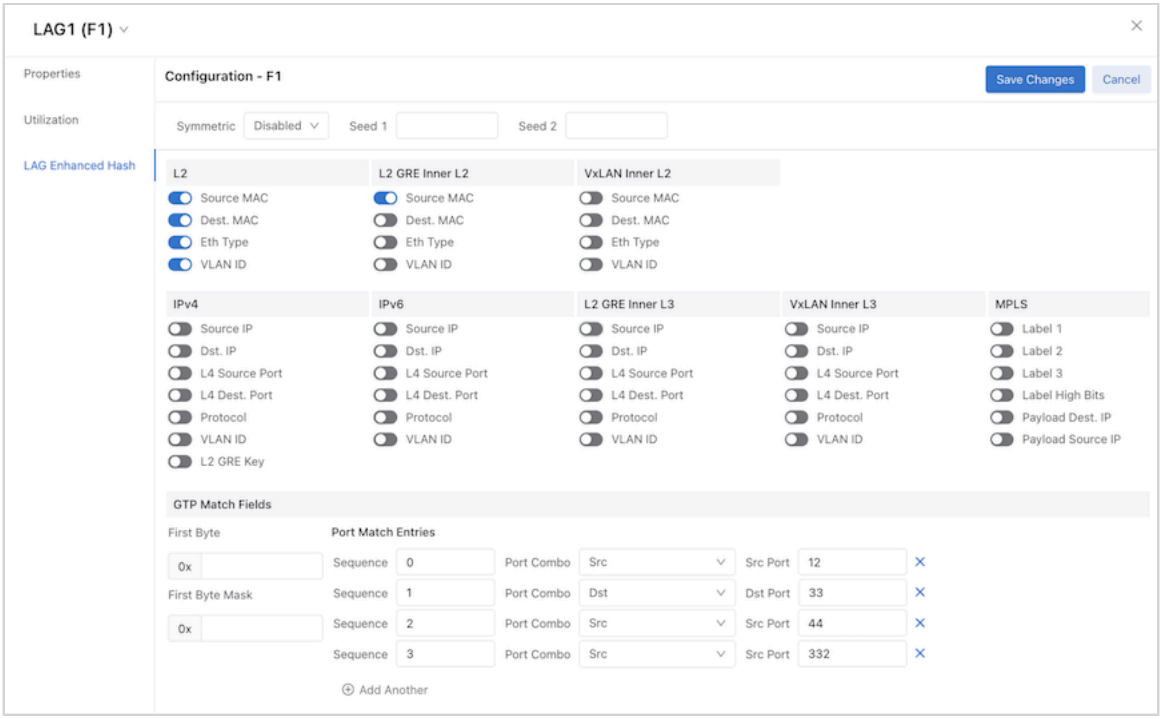
There are two time-series charts for RX and TX stats. These charts are updated every 10 seconds and display each LAG member's bit-rate / packet rate. For LAGs with over five members, select **Top 5** or **Bottom 5** from the **All** drop-down in the chart to display only the top or bottom five interfaces.

Alternate between **Bytes** and **Packets** to change the statistics displayed. Alternating between bytes and packets will not reset the time-series chart since both bytes and packet data are polled every 10 seconds.

Select **Clear/Reset LAG Stats** to re-initialize the chart. Selecting a new LAG re-initializes the charts, and polling begins for new LAG utilization data.

Enhanced Hash Configuration

Figure 7-9: Enhanced Configuration



Before the DMF 8.6 release, the LAG Enhanced Hash configuration occurred on the Switches page. Now, it is done in the **LAGs Detail** pane.

Select **Unlock to Edit** to enable changing configuration settings and **Save Changes** to submit the changes.

The LAGs detail pane closes, and the updated settings appear in the **Switch Lag Enhanced Hash Settings** grid table.

Each column lists sub-settings for **L2**, **L2 GRE Inner L2**, and other configurations.

Select a link in **Switch Lag Enhanced Hash Settings** to turn these on and off, depending on the data required. There are validations to ensure the correct combination of settings (such as **L2 GRE Inner L2** and **L2 GRE Inner L3** cannot be set at the same time) that will appear in an AlertMessageList in the pane. The configuration must pass all validations to submit the changes successfully.

The Symmetric field has three options: **Default**, **Enabled**, and **Disabled**.

- Disable the **Symmetric** field for the GTP Match settings.
- Enter **First Byte** and **First Byte Mask** as hex values 0–255 with up to four **Port Match Entries**.
- Each port match entry must have a port combo and number values for the corresponding ports.

When setting the Port Combo to **AND** or **OR**, both **Src Port** and **Dst Port** must have non-zero values.

7.1.2 L2 GRE Key Hashing

The L2 GRE Key-based hashing feature allows the L2 GRE packets to hash based on the L2 GRE (Tunnel) Key on Core DMF switches.

Previously, L2 GRE payload-based hashing (InnerL2 or InnerL3) applied only to L2 GRE packets terminated at DMF delivery or filter switches. If a user wanted to hash L2 GRE packets transiting a DMF core switch, the L2 GRE payload-based hashing across port-channel interfaces would not have been functional as the L2 GRE tunnel was not terminating on the core DMF switch.

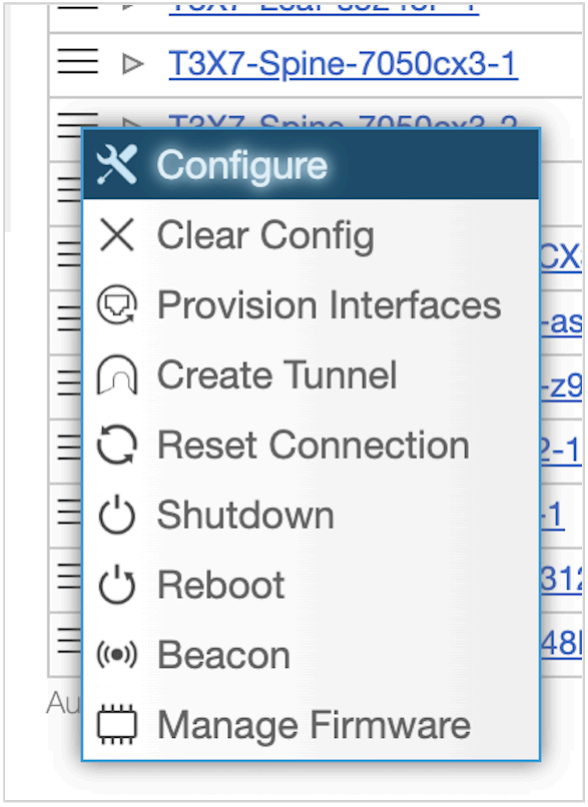
With the L2 GRE Key-based hashing feature, users can now hash L2 GRE packets based on the L2 GRE Key on core DMF switches.

Note: The L2 GRE Key-based hashing feature applies to switches running SWL OS and does not apply to switches running EOS.

Perform the following steps:

1. Configure the L2 GRE Key Hashing in the UI for a switch in the **Fabric** > **Switches** page using the table row menu action **Configure** option.

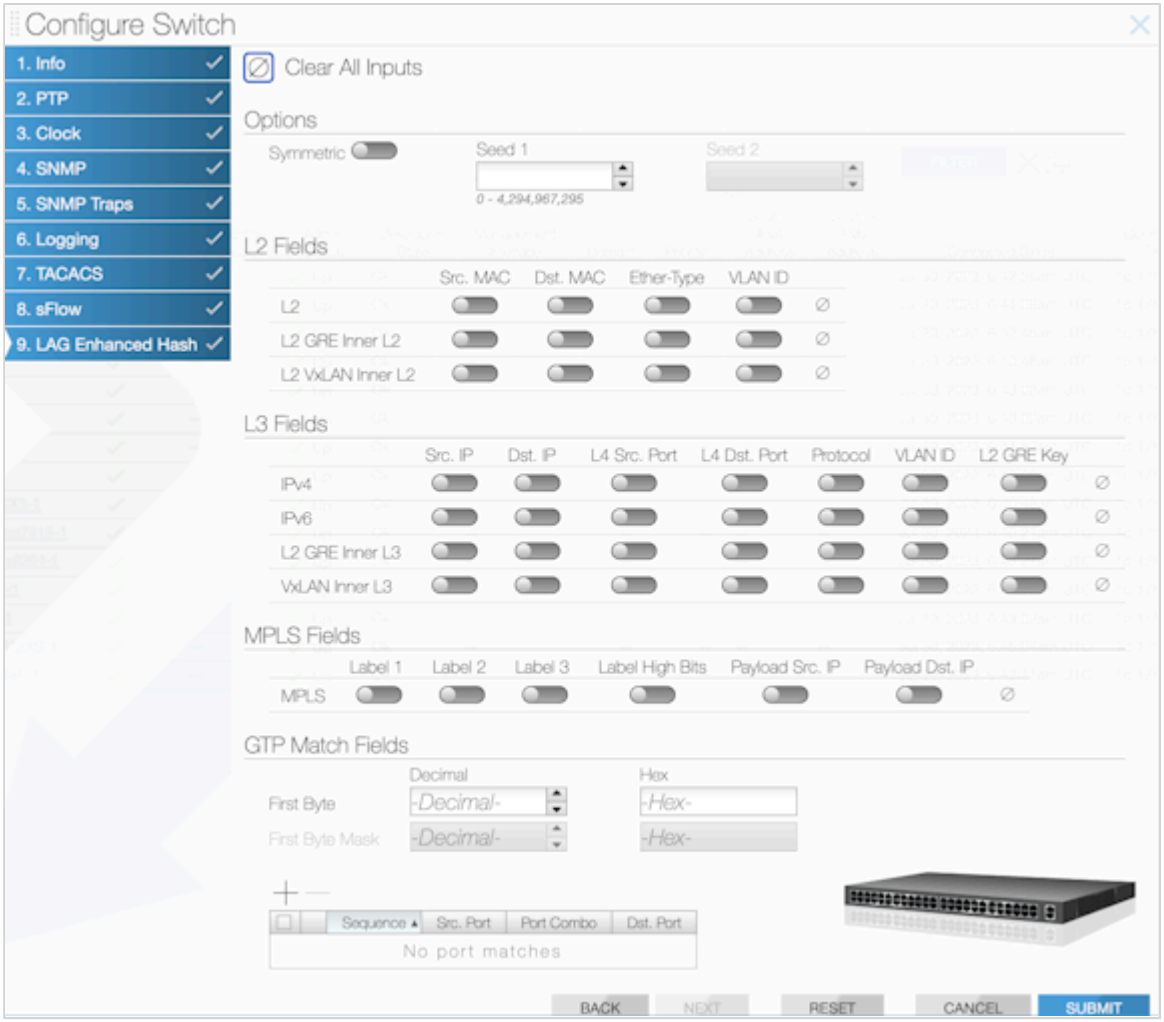
   Figure 7-10: Fabric Switch Configure Menu



2. Enable the **L2 GRE Key** for the **IPv4** packets in the **LAG Enhanced Hash** step.

Note: The L2 GRE Key is unsupported for IPv6 and VXLAN Inner L3.

Figure 7-11: Configure Switch L2 GRE Key



3. Select **Submit** to save the configuration.

Use the CLI commands to verify settings and troubleshoot any issues that may arise.

7.1.3 VXLAN Hashing

VXLAN hashing enables hashing on a VXLAN payload, including hashing on the Inner L3 Source IP, Inner L3 Destination IP, Inner L2 Source MAC, and inner L2 Destination MAC. This only applies to terminated cases.

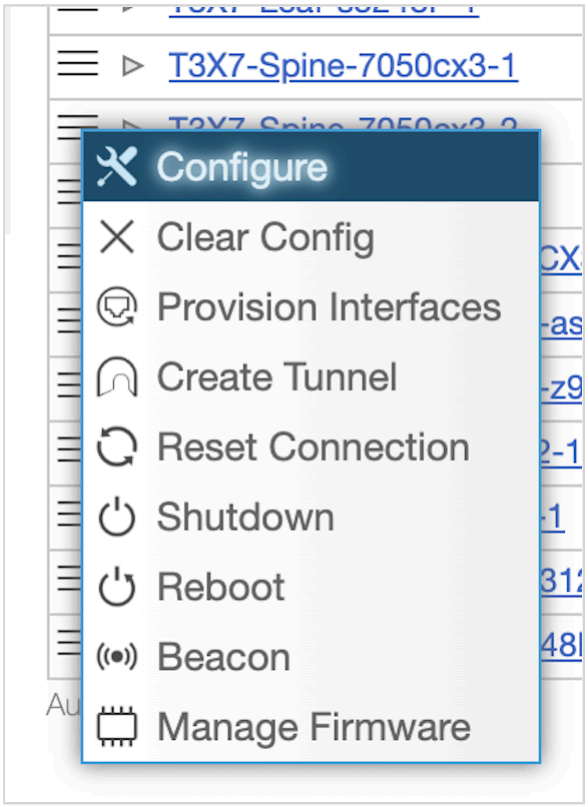
Symmetric hashing works with VXLAN packet Inner L3 Source IP/Destination IP, Inner L4 Source Port/ Destination Port, and Outer L3 Source IP/Destination IP.

Note: VXLAN hashing applies to switches running SWL OS.

Perform the following steps:

1. Configure the VXLAN Hashing in the UI for a switch in the **Fabric** > **Switches** page using the table row menu action **Configure** option.

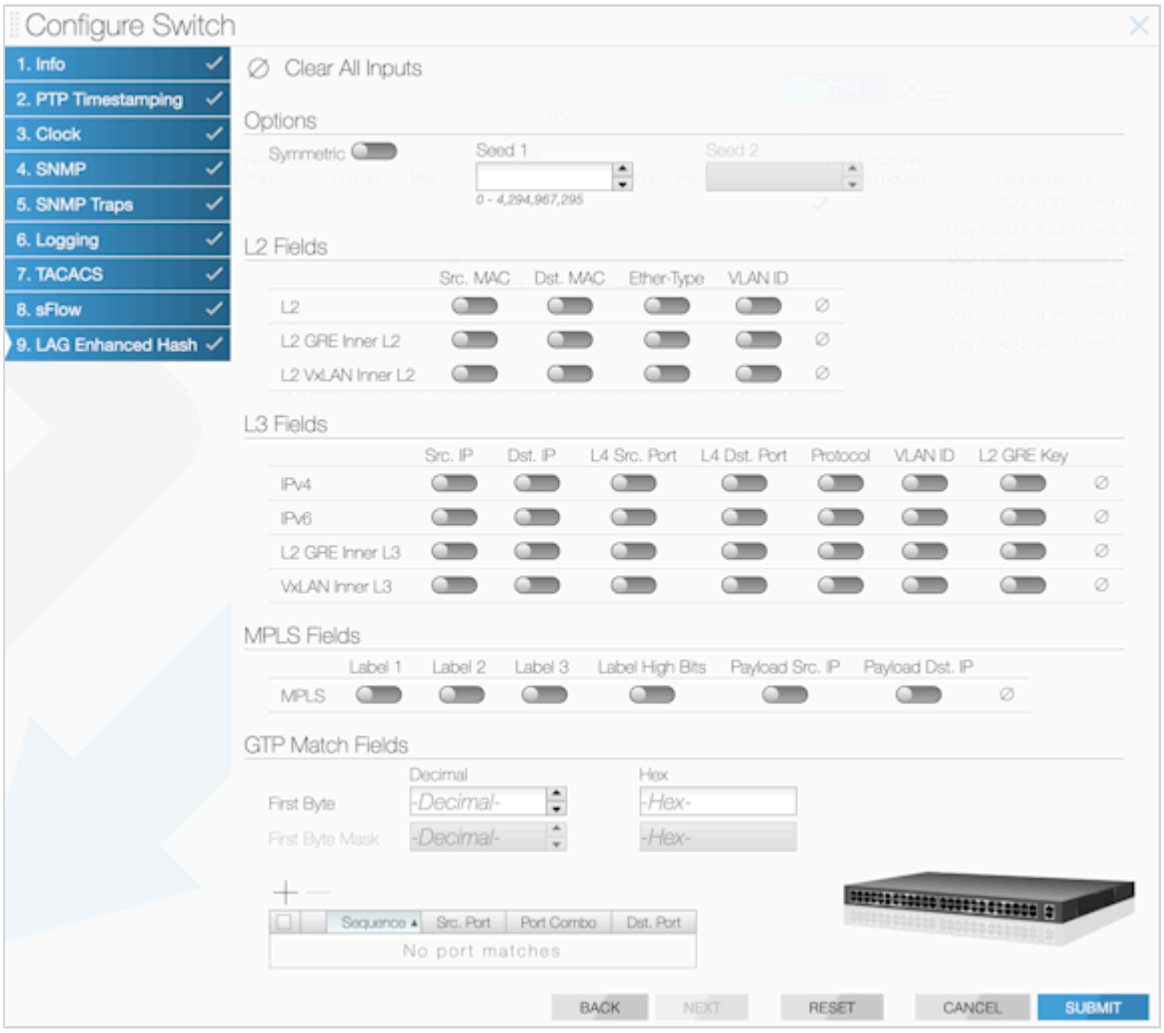
Figure 7-12: Fabric Switch Configure Menu



2. In the **LAG Enhanced Hash** step, configure the following fields depending on your requirements:
   - **L2 VXLAN Inner L2 fields**
   - **VXLAN Inner L3 fields**

> **Note:**
> - L2 GRE Key is not supported for VXLAN hash fields.
> - Cannot simultaneously specify enhanced hash for L2 GRE Inner L2 and Inner L3.
> - Cannot simultaneously specify enhanced hash for VXLAN Inner L2 and Inner L3.

Figure 7-13: Configure Switch LAG Enhanced Hash



3. Select **Submit** to save the configuration.

Use the CLI commands to verify settings and troubleshoot any issues that may arise.

7.1.4 MLAG RTAG7 Hash Computation for Unbalanced Load Balancing

In DANZ Monitoring Fabric (DMF), all SWL OS switches with the same underlying ASIC have the same RTAG7 hash parameters configured by default. After configuring MLAG to load balance traffic from filter to delivery switches, and when the number of interfaces in the MLAG and LAG in the delivery switch is the same, the traffic will not be load balanced in the delivery switch because of hash polarization.

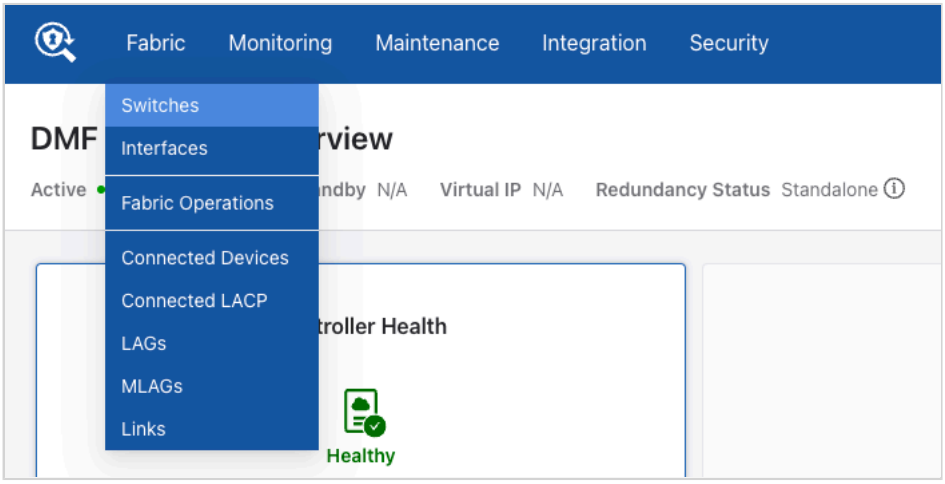
Use the feature to avoid LAG hash polarization by choosing different hash algorithms in separate switches. The other option is using a different packet field set in each switch.

The feature supports using different hash algorithms by choosing different hash seed values.

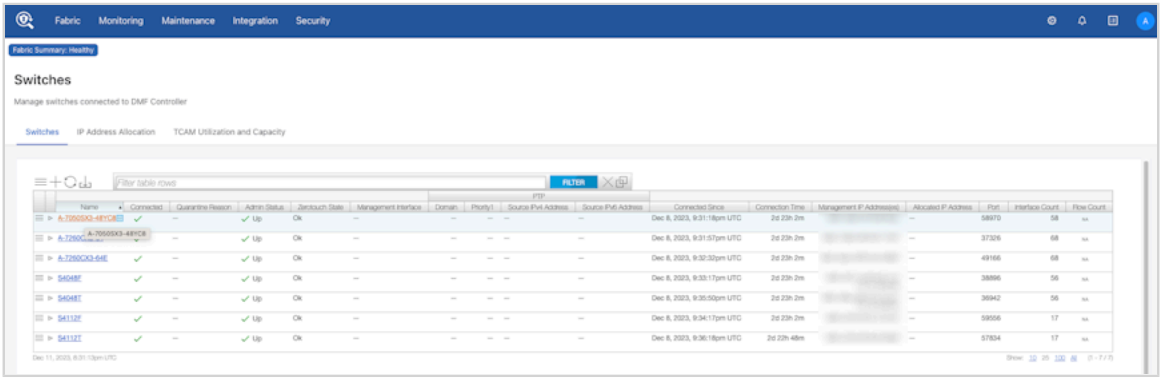
Note: MLAG RTAG7 hash computation for unbalanced load balancing only applies to switches running SWL OS, does **not** apply to switches running EOS and is only supported on select Broadcom® switch ASICs that run SWL OS.

Perform the following steps:

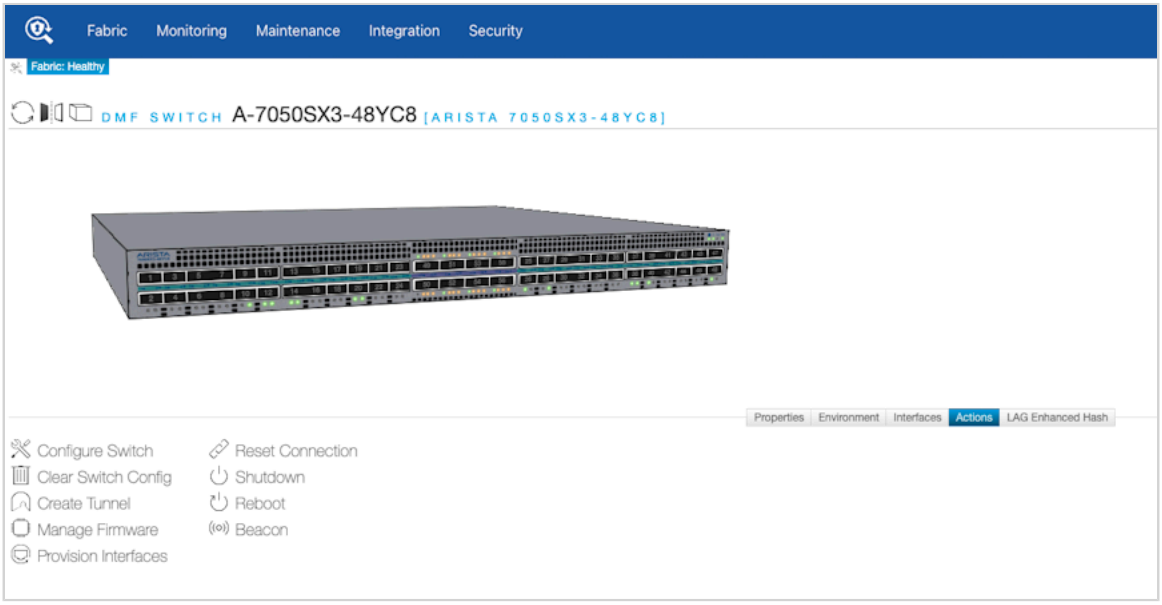
From the home page, select the switches using **Fabric > Switches**.



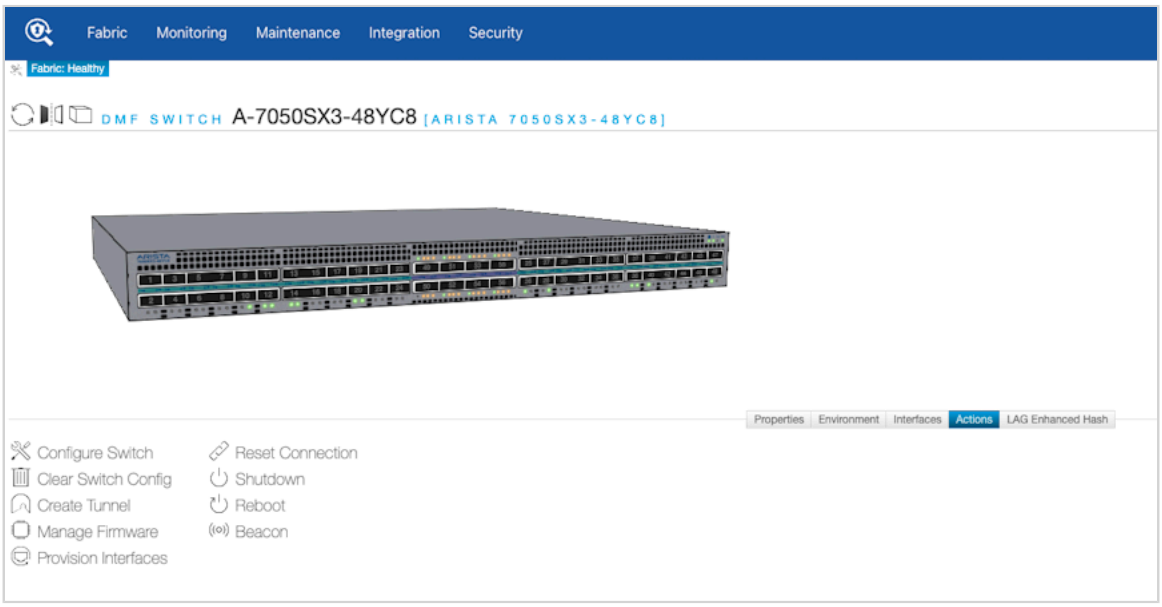
A list of available switches displays.



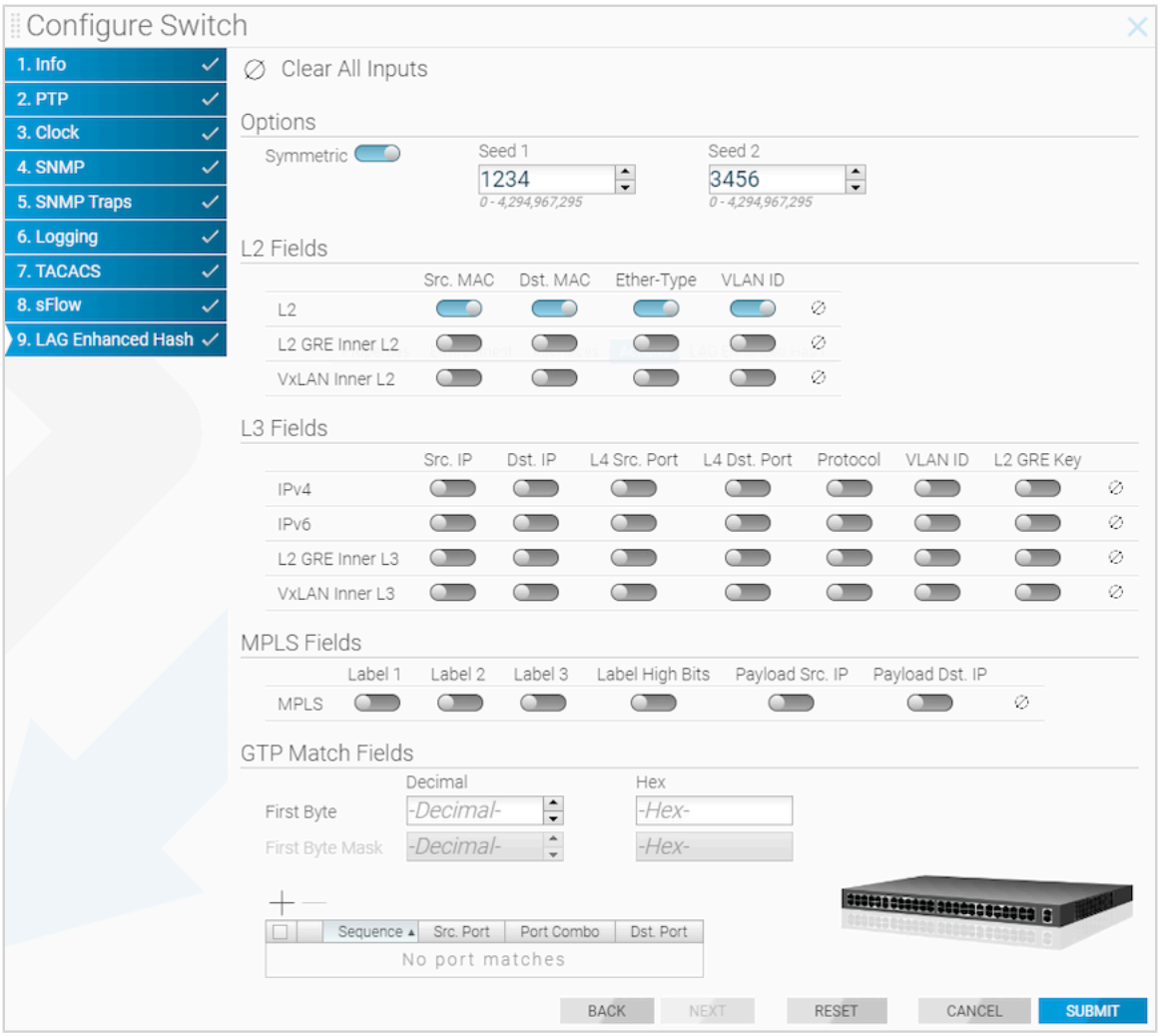
Choose the required switch for configuring the hash seeds. Hash seeds are integers used to generate random numbers in the RTAG7 hash algorithm.



Select **Actions** from the menu.



On the **Configure Switch** page, select **LAG Enhanced Hash**.



Toggle **on** the following:

- **Options**
  - **Symmetric**
- **L2 Fields**
  - **Src. MAC**
  - **Dst. MAC**
  - **Ether-Type**
  - **VLAN ID**

Enter the hash **Seed 1** and **Seed 2** values in the LAG Enhanced Hash window.

Select **Submit**.

Note: The GUI example above illustrates an **L2** configuration.

Optional: View the configured hash seed using the CLI and the following **show** command.

```
R450-C1# show lag-enhanced-hash
~~~~~~~~~~~~~~~~~~~~~~~~~~ L2 Enhanced Hash  ~~~~~~~~~~~~~~~~~~~~~~~~~~
# Switch DPID     Dst mac Eth type Src mac Vlan id Seed1 Seed2 Symmetric
-|---------------|-------|--------|-------|-------|-----|-----|---------|
1 A-7050SX3-48YC8 True    True     True    True    1234  3456  True
```

7.1.5 Trident4 LAG Hashing for L2GRE and VXLAN Traffic

Overview

For L2GRE transit traffic, LAG hashing uses only the encapsulated (inner) packet header fields. There is no option to use underlay (outer) packet header fields.

- When the encapsulated packet is IP, the system uses the IP parameters configured with **hash ipv4** or **hash ipv6** for hashing.
- When the encapsulated packet is L2, the system uses L2 parameters configured using **hash l2** for hashing.

For VXLAN transit traffic, LAG hashing uses only the underlay packet header fields. There is no option to use encapsulated packet header fields. The system treats VXLAN transit traffic like IP traffic.

This feature is supported on Arista 7050X4 switches.

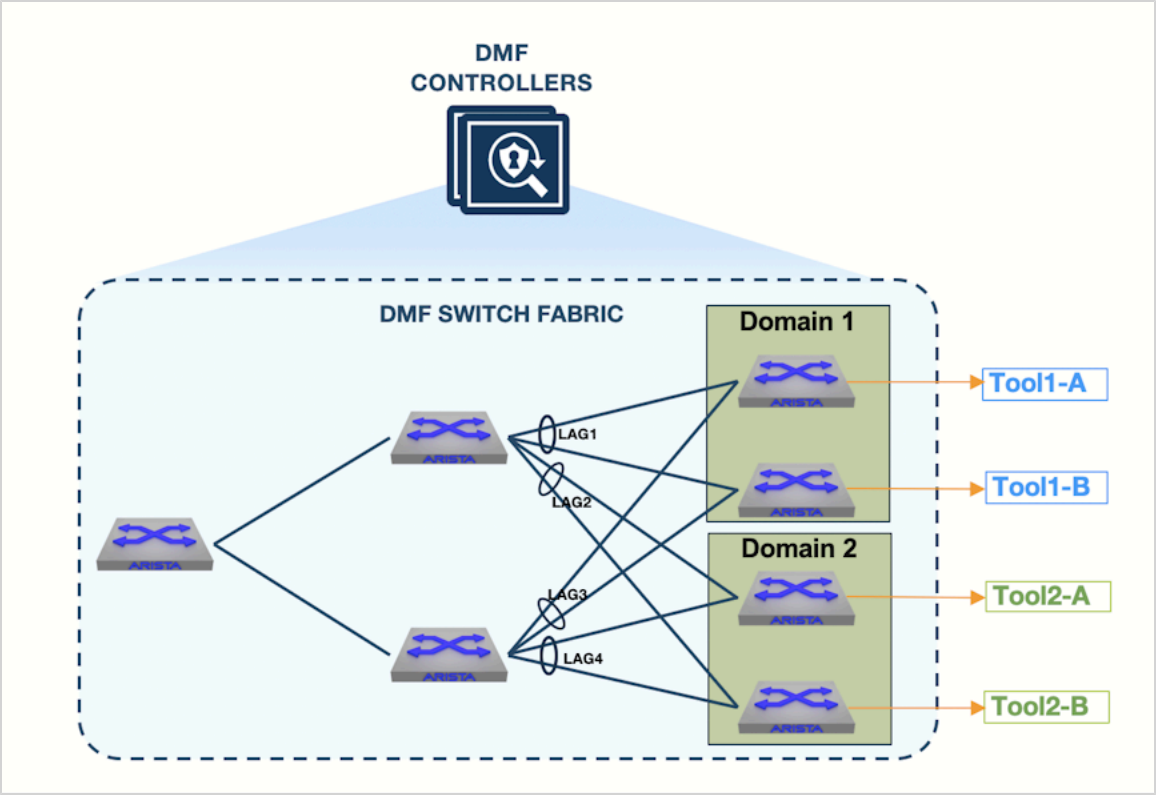
Limitations

Like other platforms, you cannot configure VXLAN and L2GRE packet fields for LAG hashing. For 7050X4 switches, the system does not support LAG configurations **hash l2gre**, **hash vxlan**, **hash ipv4**, and **l2-gre-key**.

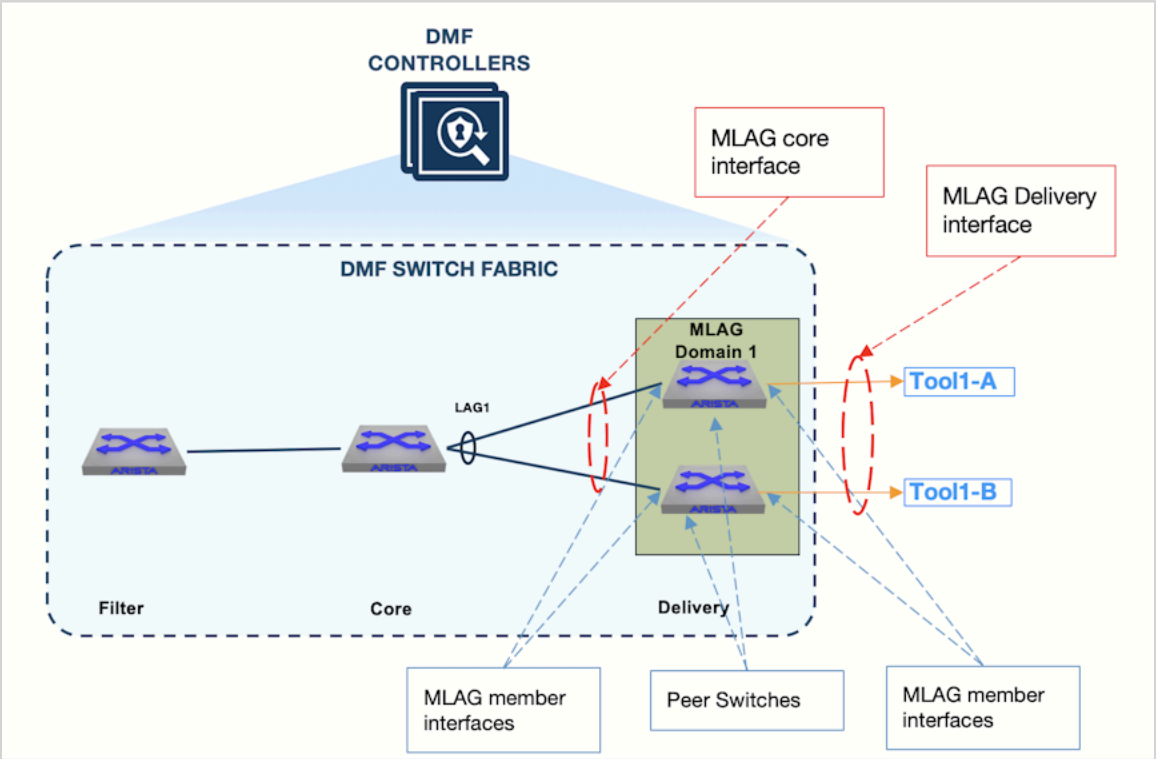
7.2 Pseudo Multi-Chassis Link Aggregation

DMF supports Link Aggregation Groups (LAGs) that allow 2 or more physical interfaces on the same DMF switch to be aggregated into 1 logical interface to increase the aggregate bandwidth and provide link redundancy against link failure. This feature works well if all the tools connect to the same DMF delivery switch, typically when co-locating customer tools in the same physical location. However, in cases where tools reside in different data centers or physical locations, where a single DMF switch cannot connect to all the tools, load balancing across two DMF delivery switches is required.

A pseudo-multi-chassis Link Aggregation Group (MLAG) provides redundancy for each delivery switch connected to a multi-homed tool. With MLAG, traffic is hashed on the upstream DMF switch across two active-active links toward the delivery switches. If one of the switches fails, the traffic will fail over to the healthy switch.



7.3 MLAG Components



- **MLAG Domain**: An MLAG domain is a logical grouping of two delivery switches that will participate in an MLAG.
- **Peer Switch**: Member switches added into the MLAG domain.
- **MLAG Interface**: An MLAG interface, configured under the MLAG domain, is a logical binding of two physical interfaces or LAG interfaces, one from each peer switch.
- **Core MLAG Link**: A fabric-facing MLAG link. A core switch LAG interface, whose members connect to the two peer switches participating in the MLAG domain.
- **Delivery MLAG Link**: An MLAG interface that is assigned the delivery interface role. This interface is used in a policy as a delivery interface.
- **MLAG Member Interface**: A physical interface or a LAG interface added into an MLAG interface.
- **DMF Policy**: A user-configured DMF policy that contains at least one MLAG delivery interface.
- **Dynamic MLAG Domain Policy**: Dynamically configured policies that follow the naming convention **`_mlag_DMF-policy_DeliverySwitch`**. For one user-configured MLAG policy, a policy that uses at least one MLAG delivery interface, two dynamic MLAG domain policies are created, one for each peer switch.

7.4 MLAG Limitations

- An MLAG domain cannot have more than two switches.
- A switch can only be a part of one MLAG domain.
- An MLAG interface can only have two member interfaces.
- An MLAG interface can only have one interface (physical interface or LAG interface) from each peer switch.
- Tunnel interfaces are not supported as members in MLAG interface configuration.

7.5 Configure MLAG

To configure an MLAG domain from the GUI, go to the **Fabric > MLAGs** tab.

Figure 7-14: Fabric –> MLAGs page

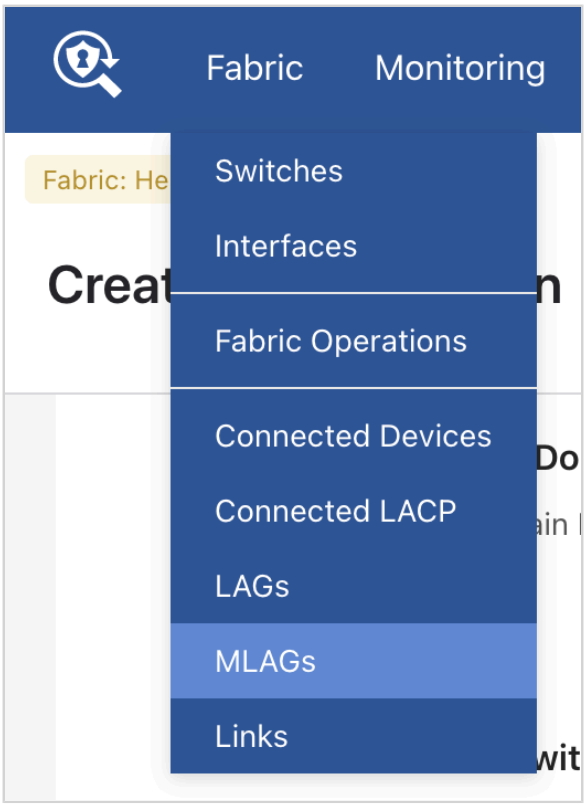
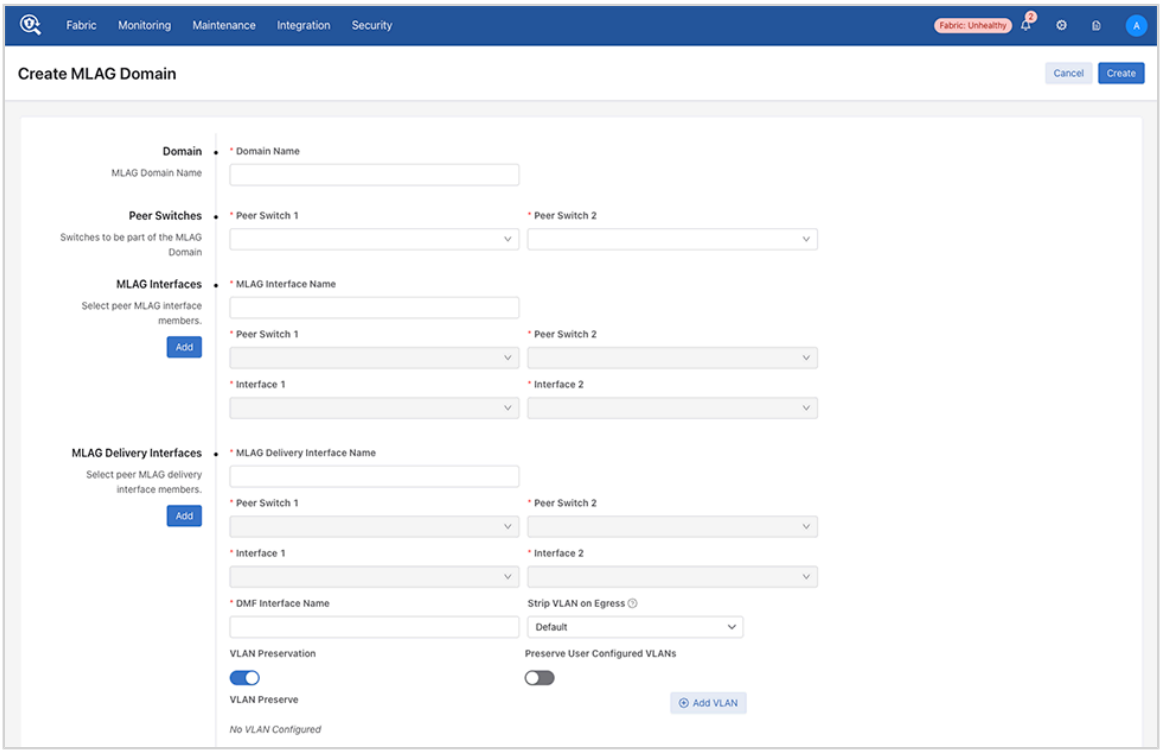


Figure 7-15: Create MLAG Domain



Select **Create MLAG Domain** and enter the following:

- **Domain Name**: Enter the MLAG domain alias.
  - **Peer Switch 1**: From the drop-down, select the first switch that will be participating in the MLAG domain.
  - **Peer Switch 2**: From the drop-down, select the second switch that will be participating in the MLAG domain.
- **MLAG Interfaces**: Enter an alias for the fabric-facing MLAG interface. This interface connects the core switch to the peer switches in the MLAG domain.
  - **Peer Switch 1 and Peer Switch 2**: After selecting peer switches under the domain name, the peer switches under the MLAG interface will automatically be selected.
  - **Interface 1**: Select the member interface that connects the core switch to ***DeliverySwitch-1***
  - **Interface 2**: Select the member interface that connects the core switch to ***DeliverySwitch-2***.
- **MLAG Delivery Interfaces**: Enter an alias for each MLAG delivery interface.
  - **DMF Interface Name**: Enter the DMF interface name for the MLAG delivery interface. This alias will be used to identify the delivery interface while configuring the DMF policy.
  - **Strip VLAN on Egress**: Select the strip VLAN configuration for the MLAG delivery interface
  - **Peer Switch 1 and Peer Switch 2**: These will be automatically selected based on the peer switches selected under the domain name.

- **Interface 1**: Select the member interface on Peer ***Switch 1***.
- **Interface 2**: Select the member interface on Peer ***Switch 2***.

Select **Create** to save the configuration.

Figure 7-16: MLAG Domain State

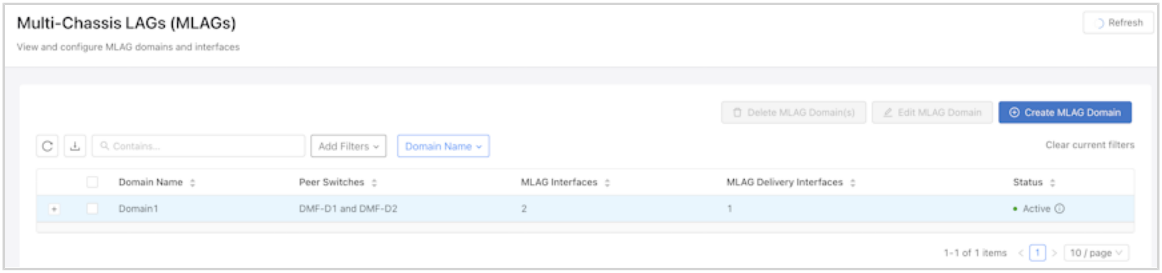
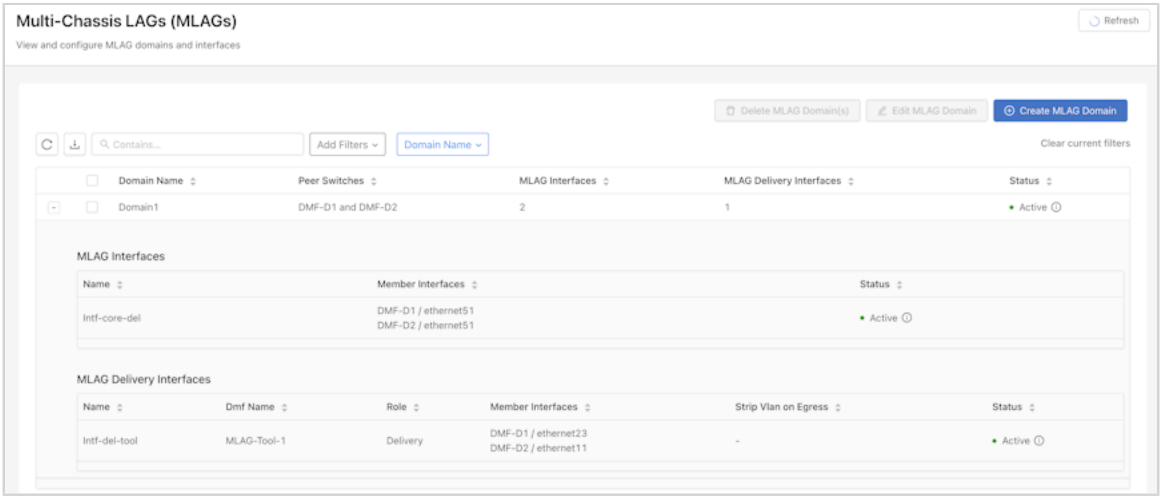


Figure 7-17: MLAG Domain Expanded View

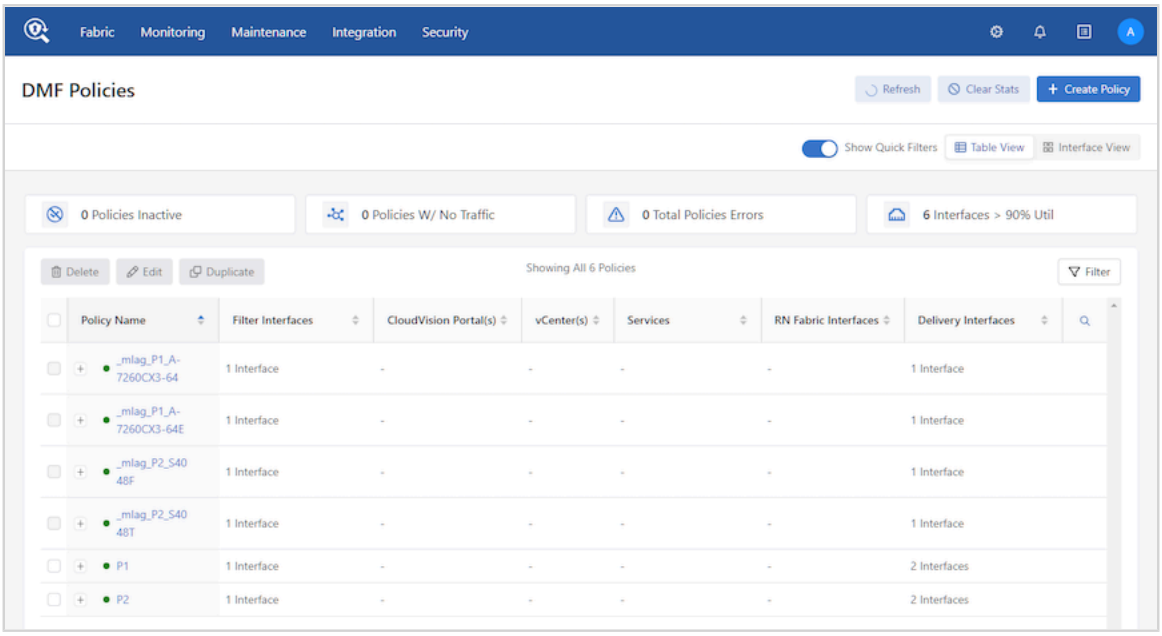


The above screenshot displays the MLAG domain status. Select **+** to expand each MLAG interface's domain configuration and status of each MLAG interface.

7.6 Create MLAG Policy

To configure an MLAG policy, go to the **Monitoring > Policies** page.

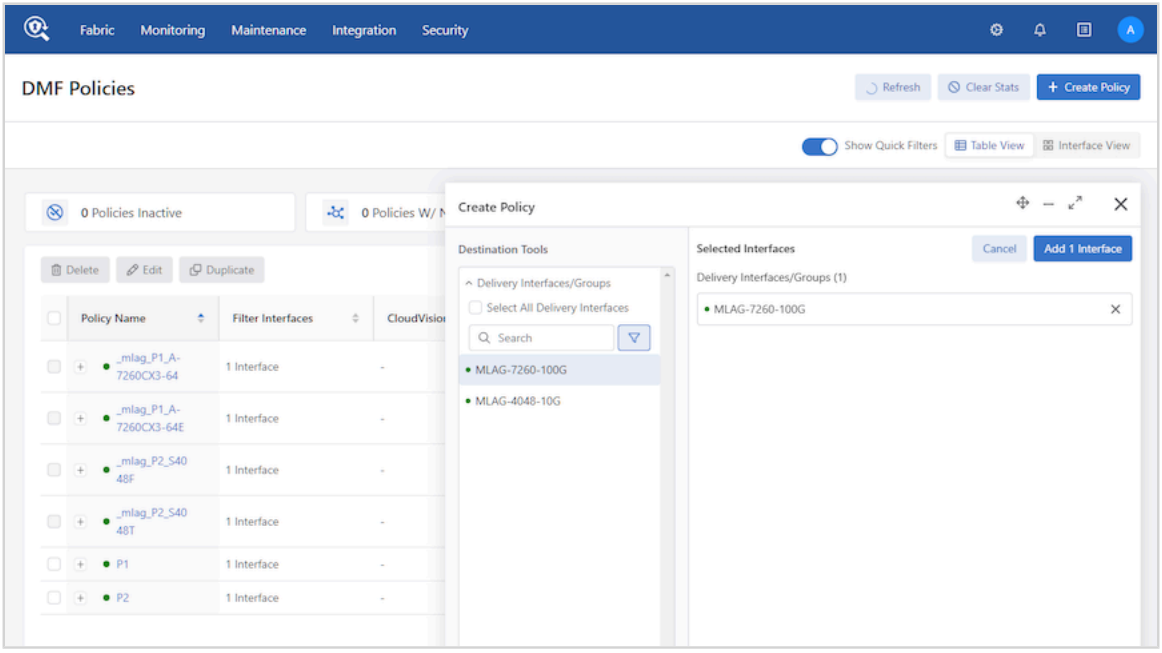
Figure 7-18: Configure MLAG Policy



Select **+ Create Policy** to add a new policy.

Select **+ Add Ports** in **Destination Tools** and select or drag the MLAG interface to associate with the policy.

Figure 7-19: Add MLAG Interface



Select **Add 1 Interface** and enter the following to configure the policy association.

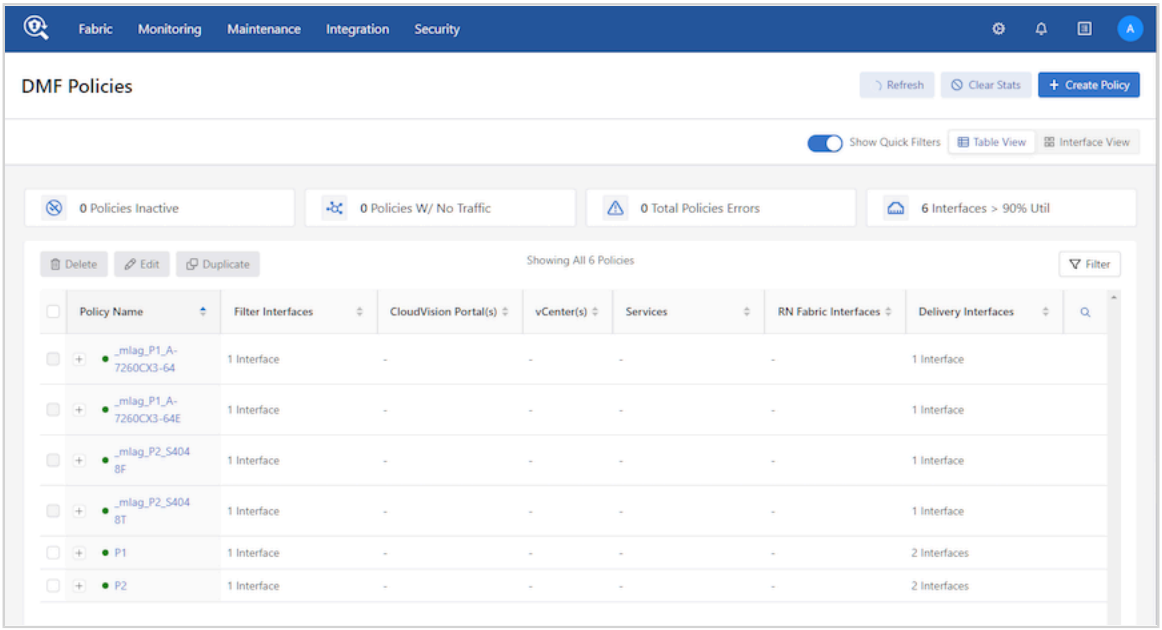
- **Name**: Assign a unique name to the policy.
- **Description**: A description for the policy.
- **Action**: **Forward** (default), **None**, **Capture**, or **Drop**.
- **Priority**: **100** (default), or enter a value.
- **Scheduling**: **Automatic** (default), **Now**, **Set Time**, or **Set Delay**.
- **Port Selection > Traffic Sources > + Add Port(s)**: Select the filter interface (traffic source) for the policy.
- **Match Traffic > Allow All Traffic / Deny All Traffic or Configure A Rule**: Specify the traffic rule for the policy.

Select **Create Policy**.

7.7 Viewing Policy Statistics

After configuring the MLAG policy, view it under **Monitoring > Policies** along with the dynamic policies created as part of the MLAG policy.

Figure 7-20: MLAG Policy



To view the policy statistics, roll over a **Policy Name** or select the **MLAG Policy Name**. The following screens appear.

Figure 7-21: MLAG Policy - Interface Statistics

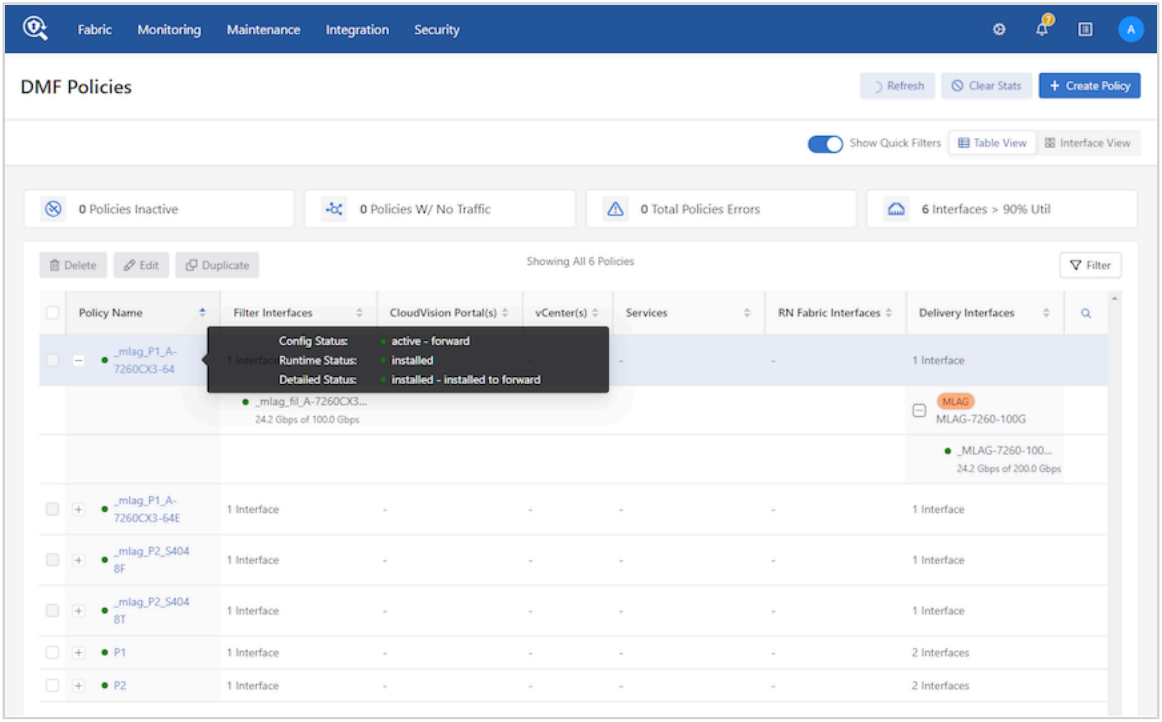
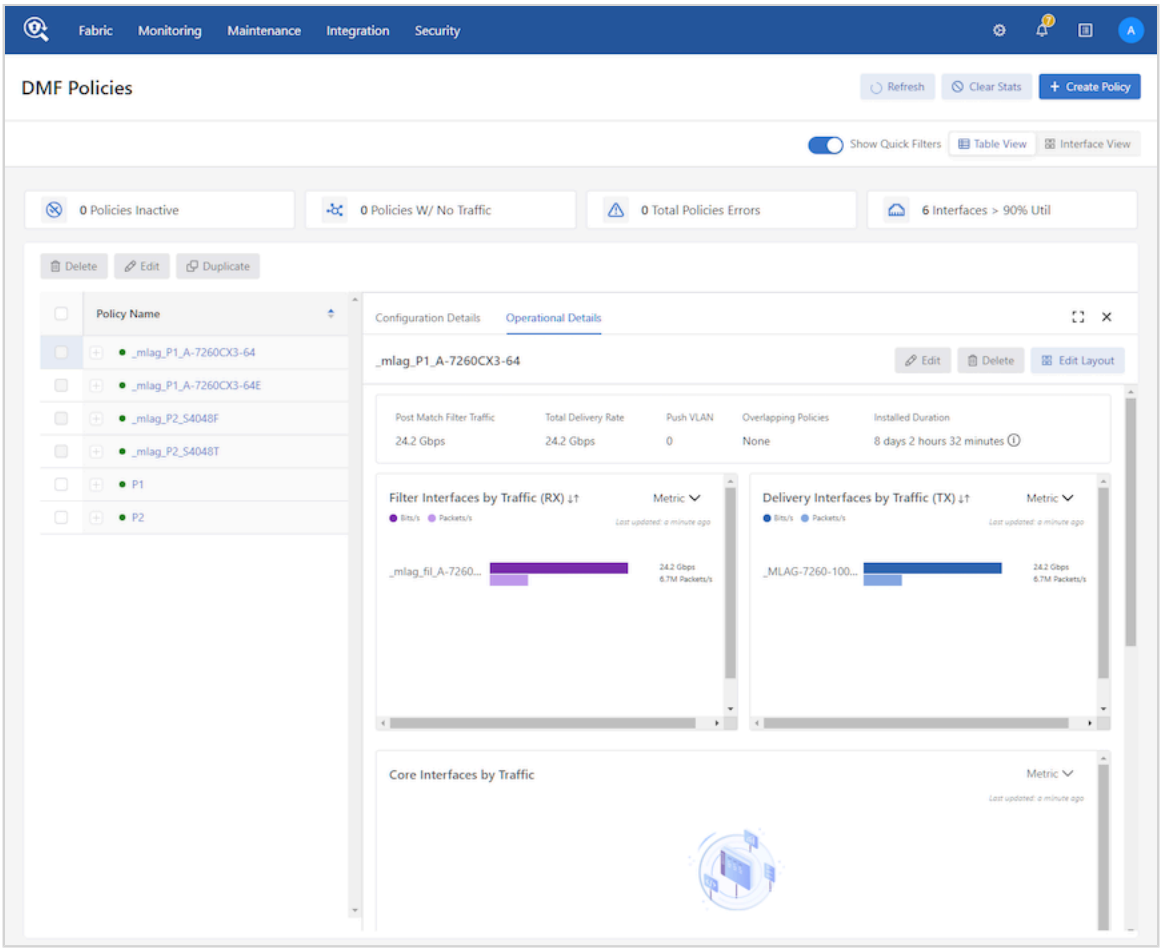


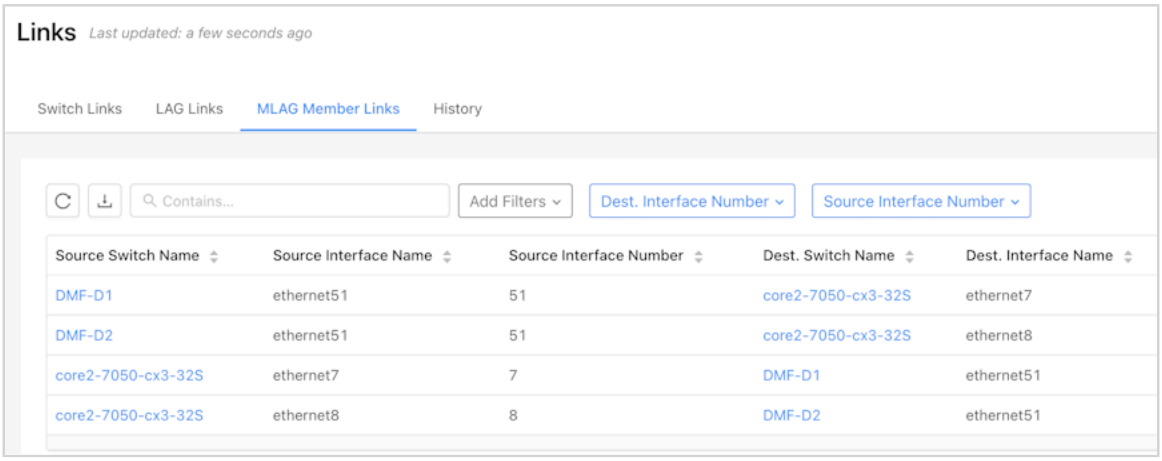
Figure 7-22: MLAG Policy - Operational Details



7.8 Viewing MLAG Links

To view the MLAG links, go to **Fabric > Links > MLAG Member Links** tab.

Figure 7-23: MLAG Member Links



The above screenshot shows the MLAG links established between the core switch and the peer switches that are part of the MLAG domain. The LLDP message exchange discovers the links.

7.9 Using the Command Line Interface

Configure Link Aggregation Groups

Configuring Hashing Fields

Minimum Link and Activation Configuration for LAG Interfaces

Tunnel Endpoint for vCenter Integration

L2 GRE Key Hashing

VXLAN Hashing

VXLAN LAG Hashing on DCS7280 Platforms

MLAG RTAG7 Hash Computation for Unbalanced Load Balancing

Configuring an MLAG

MLAG Link Discovery

Overlapping Policies in LAGs

Using LAG Interfaces as Members in MLAG Interfaces

7.9.1 Configure Link Aggregation Groups

1. Use the **lag-interface** command to enter the config-switch-lag-if submode to define the LAG member interfaces and specify the type of load distribution (hashing) to use for the LAG.
2. Use the **member** command to add an interface to a LAG. Enter this command for each interface to add to the LAG. To remove an interface, use the **no member** version of the command.

   For example, the following commands add two interfaces to a LAG named ***my-lag***.

```
controller-1(config)# switch DMF-FILTER-SWITCH-1
controller-1(config-switch)# lag-interface mylag
controller-1(config-switch-lag-if)# member ethernet13
controller-1(config-switch-lag-if)# member ethernet14
```

3. To configure multiple delivery interfaces as a LAG, complete the following steps:

   a. Assign a name to the LAG and enter the ***config-switch-lag-if*** submode.

```
controller-1(config)# switch DMF-DELIVERY-SWITCH-1
controller-1(config-switch)# lag-interface lag1
controller-1(config-switch-lag-if)#
```

   b. Assign members to the LAG.

```
controller-1(config-switch-lag-if)# member ethernet39
controller-1(config-switch-lag-if)# member ethernet40
```

4. To configure a core LAG apply the membership configuration to both ends of the connection:

```
controller-1(config)# switch SWITCH-1
controller-1(config-switch)# lag-interface core-link
controller-1(config-switch-lag-if)# member ethernet1
controller-1(config-switch-lag-if)# member ethernet3
```

```
controller-1(config)# switch SWITCH-2
controller-1(config-switch)# lag-interface core-link
controller-1(config-switch-lag-if)# member ethernet49
controller-1(config-switch-lag-if)# member ethernet51
```



Note: Ensure that when you set up a core LAG between DMF fabric switches it is properly configured on *both sides of the connection*; otherwise, you will get a `lag_misconfiguration` error state. You need to configure link aggregation symmetrically on both ends of a connection for it to work properly.

5. To view the configured LAGs, enter the **show lag** command, as in the following example:

```
controller-1> show lag
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Lag Interfaces  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Switch   LAG       State Rx Rate Pkt Rate Peak Rate Peak Pkt Rate TX Rate Pkt Rate Peak Rate Peak Pkt Rate
-|--------|---------|-----|-------|--------|---------|-------------|-------|--------|---------|-------------|
1 SWITCH-1 core-link up    720bps  0        2.59Gbps  608181        184bps  0        53.2Mbps  37150
2 SWITCH-2 core-link up    184bps  0        424bps    0             720bps  0        1.08Kbps  1

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Member Interfaces  ~~~~~~~~~~~~~~~~~~~
```

7.9.2 Configuring Hashing Fields

To configure the hashing fields manually via the CLI, use the **lag-enhanced-hash** command to enter **config-switch-hash** mode as in the following example:

```
controller1(config)# switch DMF-FILTER-SWITCH-1
controller1(config-switch)# lag-enhanced-hash
controller1(config-switch-hash)#
```

The hash commands have the following syntax.

- To hash on GTP fields, use one of the following options:

```
controller1(config-switch-hash)# hash gtp
header-first-byte Configure fields to identify GTP traffic
port-match Configure UDP tunnel port match entry
```

- To hash on IPv4 fields, use one of the following options:

```
controller1(config-switch-hash)# hash ipv4
<cr>
dst-ip Destination IPv4 address (optional)
l4-dst-port TCP/UDP destination port (optional)
l4-src-port TCP/UDP source port (optional)
protocol IP protocol (optional)
src-ip Source IPv4 address (optional)
vlan-id Vlan Id (optional)
```

- To hash on IPv6 fields, use one of the following options

```
controller1(config-switch-hash)# hash ipv6
<cr>
dst-ip Collapsed destination IPv6 address (optional)
l4-dst-port TCP/UDP destination port (optional)
l4-src-port TCP/UDP source port (optional)
nxt-hdr Next Header (optional)
src-ip Collapsed source IPv6 address (optional)
vlan-id Vlan Id (optional)
```

- To hash on Layer-2 fields, use one of the following options:

```
controller1(config-switch-hash)# hash l2
dst-mac Destination xMAC address
eth-type Ethernet Type
src-mac Source MAC address
vlan-id Vlan Id
```

- To hash on L2GRE fields, use one of the following options:

```
controller1(config-switch-hash)# hash l2gre
inner-l2 Use inner L2 fields for hash computation (optional)
inner-l3 Use inner L3 fields for hash computation (optional)
```

- To hash on MPLS labels, use one of the following options:

```
controller1(config-switch-hash)# hash mpls
<cr>
label-1 Lower 16 bits of MPLS label 1 (optional)
label-2 Lower 16 bits of MPLS label 2 (optional)
label-3 Lower 16 bits of MPLS label 3 (optional)
label-hi-bits Higher 4 bits of MPLS Labels 1,2 and 3 (optional)
```

- To manually configure the hash seeds:

```
controller1(config-switch)# hash seeds
<First hash seed> Configure seed1 for hash computation
controller1(config-switch-hash)# hash seeds 3809
<cr>
<Second hash seed> Configure seed2 for hash computation (optional)
controller1(config-switch-hash)# hash seeds 3809 90901
<cr>
```

- To enable or disable symmetric hashing:

```
controller1(config-switch-hash)# hash symmetric
<cr>
disable Disable symmetric hashing
enable Enable symmetric hashing
```

7.9.3 Minimum Link and Activation Configuration for LAG Interfaces

When some LAG member links go down, it may be preferable to isolate the filter switch by bringing down the entire LAG interface rather than delivering unreliable data to tools and devices.

Two additional commands are part of the DANZ Monitoring Fabric (DMF) **lag-interface** configuration to aid in managing the LAG interface when a specified number of links go down. These commands are:

- **minimum-link**
- **activate**

minimum-link

The **minimum-link** command configures the minimum number of links that must be up for a functional LAG. If the total number of active links in the **interface-group** is less than the configured minimum threshold, DMF brings down all active links belonging to the interface group.

The **minimum-link** command is optional, with a default value of **0**. You should ensure this value is reasonable. If the value exceeds the total member links, the LAG will always be down.

activate

You must use the **activate** command to manually try to bring up the LAG if it was shut down because the number of active links was below the minimum link.



Note: If you make any changes to fix the downed LAG members and subsequently use the **activate** command to re-enable the LAG, it will go back down if the active members are still less than the minimum links value.

Configuration

The **minimum-link** and **activate** commands are available within a switch's lag-interface submode. The following example illustrates configuring these options:

Use the CLI to enter the switch's submode.

1. Enter the lag interface submode by using the **lag-interface** command.
2. Use the **minimum-link** command to configure the minimum threshold value needed before bringing down a LAG.
3. Use the **activate** command to bring back up the previously shutdown LAG (due to having fewer active member links than the minimum threshold).

```
dmf-controller-1(config)# switch s1
dmf-controller-1(config-switch)# lag-interface lag1
dmf-controller-1(config-switch-lag-if)# minimum-link 2
dmf-controller-1(config-switch-lag-if)# activate
```

Show Commands

While in the **switch** and **lag-interface** submodes, the **show running-config** and **show this** commands list the configured **minimum-link** threshold value. However, since **activate** is an action command rather than a configuration value, it doesn't appear when running the `show` commands.

```
dmf-controller-1# show running-config switch core1 lag lag1
! switch
switch core1
 !
 lag-interface lag1
 member a
 member b
 member c
 member d
 minimum-link 2
```

Troubleshooting

- If a LAG always shows down, ensure the **minimum-link** value is not greater than the total number of member links in that LAG.

Limitations

- The system doesn’t provide any warning when the **minimum-link** value exceeds the total number of member links in that LAG.

7.9.4 Tunnel Endpoint for vCenter Integration

This feature supports using Link Aggregation Group (LAG) in the tunnel endpoint configuration and runs on DANZ Monitoring Fabric (DMF) compatible switches supporting LAGs. Refer to the DMF 8.6 Hardware Compatibility Guide to view the list of compatible switches.

Configuration

Add the configured LAG as an interface in the tunnel endpoint configuration, as shown in the following example.

```
dmf-controller-1(conf)# tunnel-endpoint tep1 switch swl-leaf-RU35 lag1 ip-address 192.168.199.254
 mask 255.255.255.0 gateway 192.168.199.1
```

Show Commands

Review the tunnel endpoint configuration using the following command:

```
dmf-controller-1# show running-config tunnel-endpoint

! tunnel-endpoint
tunnel-endpoint tep1 switch swl-leaf-RU35 lag1 ip-address 192.168.199.254 mask 255.255.255.0
 gateway 192.168.199.1
```

Limitations

- Do not use LAG members as a Tunnel endpoint interface.
- Do not configure multiple gateways for the same interface used in Tunnel endpoints.
- Do not use a LAG configured as management or DMF interface (filter or delivery) in the Tunnel endpoint configuration.

7.9.5 L2 GRE Key Hashing

L2 GRE Key-based hashing is supported only for the IPv4-based packets with L2 GRE payload. This feature does NOT support the IPv6 packets with L2 GRE payloads.

Enable the L2 GRE Key hashing by setting the **`l2-gre-key`** parameter, as shown in the following example.

```
Controller-Active# show running-config switch DMF-SWITCH-1
! switch
switch DMF-SWITCH-1
mac c0:d6:82:17:fd:5a
!
lag-enhanced-hash
hash ipv4 l2-gre-key
hash symmetric disable
```

Use the CLI commands to verify settings and troubleshoot any issues that may arise.

```
# show lag-enhanced-hash
```

While logged into a switch, use the following commands to troubleshoot this feature.

```
root@DMF-SWITCH-1:~# ofad-ctl gt PORT_CHANNEL_ENHANCED_HASH_FIELD
Hash Field Configs:
-------------------
Symmetric Hashing:
Disabled
L2GRE Key Hashing:
Enabled
L2 Fields:
IPv4 Fields:
DSTL4 SRCL4
IPv6 Fields:
MPLS Fields:
L2GRE L2 Fields:
L2GRE L3 Fields:
VXLAN L2 Fields:
VXLAN L3 Fields:

root@DMF-SWITCH-1:~# ofad-ctl bshell getreg RTAG7_HASH_CONTROL_L2GRE_MASK_A
RTAG7_HASH_CONTROL_L2GRE_MASK_A.ipipe0[1][0x6a001900]=0xffffffff
: <L2GRE_TUNNEL_GRE_KEY_MASK_A=0xffffffff>

root@DMF-SWITCH-1:~# ofad-ctl bshell getreg RTAG7_HASH_CONTROL_L2GRE_MASK_B
RTAG7_HASH_CONTROL_L2GRE_MASK_B.ipipe0[1][0x6a001a00]=0xffffffff
: <L2GRE_TUNNEL_GRE_KEY_MASK_B=0xffffffff>

root@s5248f-1:~# ofad-ctl bshell getreg RTAG7_L2GRE_PAYLOAD_L2_HASH_FIELD_BMAP
RTAG7_L2GRE_PAYLOAD_L2_HASH_FIELD_BMAP.ipipe0[1][0x6a001b00]=0: <
L2GRE_PAYLOAD_L2_BITMAP_B=0,L2GRE_PAYLOAD_L2_BITMAP_A=0>

root@s5248f-1:~# ofad-ctl bshell getreg RTAG7_L2GRE_PAYLOAD_L3_HASH_FIELD_BMAP
RTAG7_L2GRE_PAYLOAD_L3_HASH_FIELD_BMAP.ipipe0[1][0x6a001c00]=0: <
L2GRE_PAYLOAD_L3_BITMAP_B=0,L2GRE_PAYLOAD_L3_BITMAP_A=0>
root@DMF-SWITCH-1:~#
```

Note: The L2GRE_KEY offset is the same as the SRCL4 and DSTL4 offset in hardware. Hence, the hardware requires setting SRCL4 and DSTL4 hash fields and the L2GRE_KEY hash field to hash the packets using the L2GRE_KEY.

7.9.6 VXLAN Hashing

VXLAN hashing enables hashing on a VXLAN payload, including hashing on the Inner L3 Source IP, Inner L3 Destination IP, Inner L2 Source MAC, and inner L2 Destination MAC. This only applies to terminated cases.

Symmetric hashing works with VXLAN packet Inner L3 Source IP/Destination IP, Inner L4 Source Port/ Destination Port, and Outer L3 Source IP/Destination IP.

Note: VXLAN hashing applies to switches running SWL OS.

VXLAN hashing includes hashing on L2 and L3 and the setting of at least one parameter enabled under the switch construct on Controller CLI:

```
# lag-enhanced-hash
hash vxlan inner-l2 dst-mac
hash vxlan inner-l3 dst-ip
```

CLI Commands

Use the following CLI commands to verify settings and troubleshoot any issues that may arise.

```
# show lag-enhanced-hash
```

Use the following commands to troubleshoot this feature. For example, when the hashing happens on VXSLAN payload inner L3 Src IP.

```
root@DMF-SWITCH-1:~# root@mrv1:~# ofad-ctl gt PORT_CHANNEL_ENHANCED_HASH_FIELD
Hash Field Configs:
-------------------
Symmetric Hashing:
Disabled
L2 Fields:
IPv4 Fields:
IPv6 Fields:
MPLS Fields:
L2GRE L2 Fields:
L2GRE L3 Fields:
VXLAN L2 Fields:
VXLAN L3 Fields:
IP4SRC_LO IP4SRC_HI

root@mrv1:~# ofad-ctl bshell getreg RTAG7_HASH_CONTROL_4
RTAG7_HASH_CONTROL_4.ipipe0[1][0x6a000700]=3:
<VXLAN_PAYLOAD_HASH_SELECT_B=1,VXLAN_PAYLOAD_HASH_SELECT_A=1,DISABLE_HASH_VXLAN_B=0,DISABLE_HASH
_VXLAN_A=0>

root@mrv1:~# ofad-ctl bshell getreg RTAG7_VXLAN_PAYLOAD_L2_HASH_FIELD_BMAP
RTAG7_VXLAN_PAYLOAD_L2_HASH_FIELD_BMAP.ipipe0[1][0x6a001d00]=0: <
VXLAN_PAYLOAD_L2_BITMAP_B=0,VXLAN_PAYLOAD_L2_BITMAP_A=0>

root@mrv1:~# ofad-ctl bshell getreg RTAG7_VXLAN_PAYLOAD_L3_HASH_FIELD_BMAP
RTAG7_VXLAN_PAYLOAD_L3_HASH_FIELD_BMAP.ipipe0[1][0x6a001e00]=0x1800c00
: <VXLAN_PAYLOAD_L3_BITMAP_B=0xc00,VXLAN_PAYLOAD_L3_BITMAP_A=0xc00>
```

7.9.7 VXLAN LAG Hashing on DCS7280 Platforms

The following information addresses Virtual Extensible LAN (VXLAN) hashing capabilities and behavior, specifically on the DCS-7280 platforms.

Platform Compatibility

There are two use cases:

1. **Decapsulation** – Strips the VXLAN header.
2. **Transit** – Retains the VXLAN header.

Since VXLAN Header Stripping is only supported on DCS-7280R3 platforms, the **Decapsulation** use case is supported only on these platforms.

The **Transit** use case is supported on all DCS-7280 platforms.

Configuration

There are two assumptions:

1. The VXLAN packet is IPv4 (outer).
2. The VXLAN packet has a UDP destination port value matching the configured value (default – 4789).

For the decapsulation use case, the packet context is advanced to the start of the payload upon VXLAN parsing on ingress. Hence, you can configure normal (outer) hash fields to hash against the VXLAN payload (inner) as if the outer encapsulation has already been discarded.

Traffic is hashed based on the outer header for the transit use case, just like any non-VXLAN UDP packet. Arista advises configuring L4-src-port hashing to hash against the UDP source port of the VXLAN packet, typically used as an entropy of its payload.

```
> enable
# config
(config)# switch switch-name
(config-switch)# lag-enhanced-hash
```

Limitations

- Only IPv4 VXLAN tunnel header stripping is supported.
- Currently, it is not possible to limit VXLAN packet parsing to selected interfaces on a given switch. Configuring **strip-vxlan** on any switch interface will trigger VXLAN packet parsing to be active globally on the switch. In this situation, the packet context of any ingressed VXLAN packet would be advanced to the start of its payload. As a result, policy matching and hashing behaviors will be generally affected for VXLAN packets on this switch, even for packets not subject to **strip-vxlan**.

7.9.8 MLAG RTAG7 Hash Computation for Unbalanced Load Balancing

In DANZ Monitoring Fabric (DMF), all SWL OS switches with the same underlying ASIC have the same RTAG7 hash parameters configured by default. After configuring MLAG to load balance traffic from filter to delivery switches, and when the number of interfaces in the MLAG and LAG in the delivery switch is the same, the traffic will not be load balanced in the delivery switch because of hash polarization.

Use the feature to avoid LAG hash polarization by choosing different hash algorithms in separate switches. The other option is using a different packet field set in each switch.

The feature supports using different hash algorithms by choosing different hash seed values.

Note: MLAG RTAG7 hash computation for unbalanced load balancing only applies to switches running SWL OS, does **not** apply to switches running EOS and is only supported on select Broadcom® switch ASICs that run SWL OS.

CLI Configuration

Configure the feature on each switch using the following CLI commands:

```
(config)# switch S4048T
(config-switch)# lag-enhanced-hash
(config-switch)# hash symmetric enable
(config-switch-hash)# hash seeds 1234 3456
(config-switch-hash)# hash l2 dst-mac
(config-switch-hash)# hash l2 src-mac
(config-switch-hash)# hash l2 eth-type
(config-switch-hash)# hash l2 vlan-id
```

Use the following **show** command to view the configured hash seed.

```
R450-C1# show lag-enhanced-hash switch S4048T
~~~~~~~~~~~~~~~~~~~~~~~~~ L2 Enhanced Hash  ~~~~~~~~~~~~~~~~~~~~~~~~~
# Switch DPID Dst mac Eth type Src mac Vlan id Seed1 Seed2 Symmetric
-|-----------|-------|--------|-------|-------|-----|-----|---------|
1 S4048T      True    True     True    True    1234  3456  True
```

Note: The earlier CLI example illustrates an **L2** configuration.

7.9.8.1 Troubleshooting a Configured Hash Seed

Use the following procedure to ensure the configured hash seed is correctly applied on the switch.

1. Log in to the switch using the **`connect switch switch_name`** command.
2. Use the **`ofad-ctl gt PORT_CHANNEL_ENHANCED_HASH_SEED`** command to print the seed values and the hash algorithm.
3. Ensure the hash algorithms differ in the switches where hash polarization occurs.
4. Configure the hash seeds such that the hash algorithms are different.

Example

```
R450-C1(config)# connect switch A-7050SX3-48YC8
Switch Light OS SWL-OS-DMF-8.6.x(0), 2023-12-07.09:17-42d8658
Linux A-7050SX3-48YC8 4.19.296-OpenNetworkLinux #1 SMP Thu Dec 7 09:28:30 UTC 2023 x86_64
Switch Light ZTN Manual Configuration. Type help or ? to list commands.
(ztn-config) debug bash
****************************  WARNING  ****************************
Any/All activities within bash mode are UNSUPPORTED
This is intended ONLY for additional debugging ONLY by Arista TAC.
Please type "exit" or Ctrl-D to return to the CLI
****************************  WARNING  ****************************
root@A-7050SX3-48YC8:~# ofad-ctl gt PORT_CHANNEL_ENHANCED_HASH_SEED
GENTABLE : port_channel_enhanced_hash_seed
GENTABLE ID : 0x0006
hash_seed_valid: true
HASH_SEED1=0x000004d2
HASH_SEED2=0x000004d2
computed hash algorithm from seed: CRC32LO
root@A-7050SX3-48YC8:~#
```

7.9.9 Configuring an MLAG

To configure an MLAG, use the following steps:

1. Configure an MLAG domain by specifying an alias, and add peer switches that will be participating in the MLAG.

```
Controller-1(config)# mlag-domain MLAG-Domain1
Controller-1(config-mlag-domain)# peer-switch DeliverySwitch-1
Controller-1(config-mlag-domain)# peer-switch DeliverySwitch-2
```

2. Configure the core MLAG interface.

```
Controller-1(config-mlag-domain)# mlag-interface MLAG-Core-Intf
Controller-1(config-mlag-domain-if)# member switch DeliverySwitch-1 interface ethernet50
Controller-1(config-mlag-domain-if)# member switch DeliverySwitch-2 interface ethernet50
```

   The above MLAG interface configuration selects one physical interface from each peer switch added into the MLAG domain. This MLAG interface is fabric-facing, which means that ***ethernet50*** of ***DeliverySwitch-1*** and ***ethernet50*** of ***DeliverySwitch-2*** are connected to the DMF core switch, where traffic hashing is performed.

3. Configure the core LAG interface, a LAG interface on the core switch. The members of the LAG interface are connected to the peer switches in the MLAG domain. This configuration ensures that the traffic will be hashed toward the two connected delivery switches.

```
Controller-1(config)# switch CoreSwitch-1
Controller-1(config-switch)# lag-interface Core-LAG
Controller-1(config-switch-lag-if)# member ethernet10
Controller-1(config-switch-lag-if)# member ethernet20
```

4. Configure the delivery MLAG interface by specifying an interface alias and selecting one member from each delivery switch.

```
Controller-1(config-mlag-domain-if)# mlag-interface MLAG-Del-Intf
Controller-1(config-mlag-domain-if)# member switch DeliverySwitch-1 interface ethernet1
Controller-1(config-mlag-domain-if)# member switch DeliverySwitch-2 interface ethernet1
Controller-1(config-mlag-domain-if)# role delivery interface-name MLAG-Tool-1
```

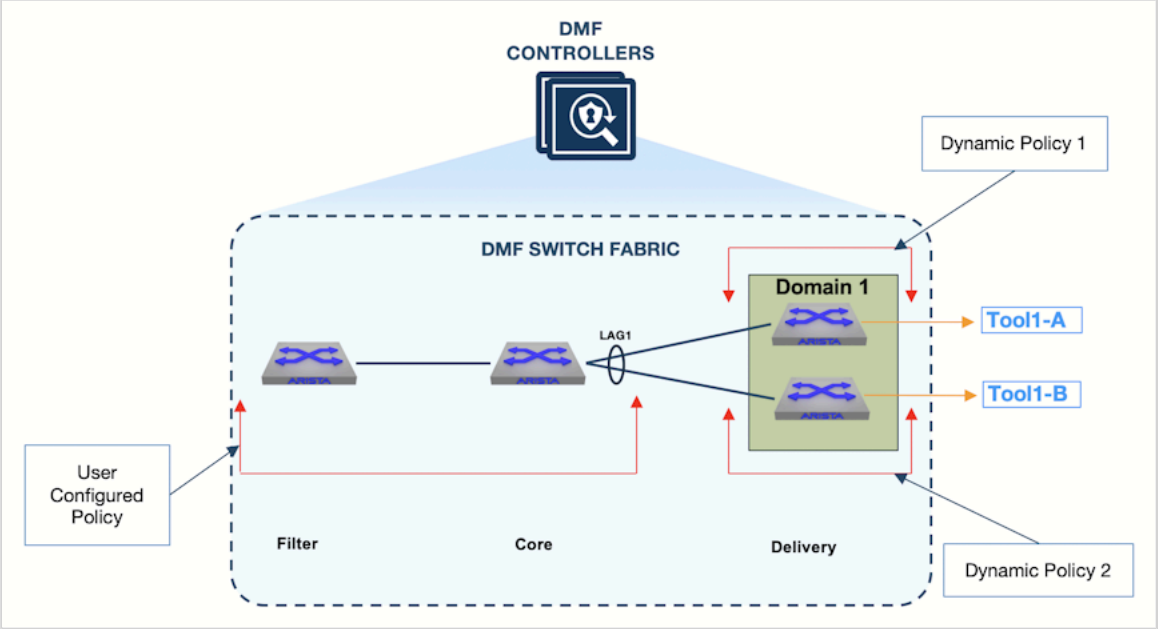
   The above MLAG interface configuration selects one physical interface from each peer switch added into the MLAG domain. The members of this MLAG interface, ***ethernet1*** of ***DeliverySwitch-1*** and ***ethernet1*** of ***DeliverySwitch-2***, are connected to multi-homed tools. Note that unlike the core MLAG interface, the delivery MLAG interface is assigned the delivery role and its interface name is configured, so that it can be used in DMF policies as a delivery interface.

5. Configure a DMF policy by following the procedure shown below:

```
Controller-1(config)# policy Policy-1
Controller-1(config-policy)# action forward
Controller-1(config-policy)# 1 match any
Controller-1(config-policy)# filter-interface Filter-1
Controller-1(config-policy)# delivery-interface MLAG-Tool-1
```

The above policy is configured using the ***MLAG-Tool-1*** interface configured in **Step 4**. Configuring the policy to use an MLAG delivery interface will result in two dynamic policies, one for each peer switch. Refer to the following topology for the policy breakdown.

Figure 7-24: MLAG Policy Breakdown



As seen in the topology above:

- The user-configured policy delivers traffic from the filter switch to the core switch LAG interface.
- Dynamic Policy 1 delivers traffic to delivery ***switch 1***.
- Dynamic Policy 2 delivers traffic to delivery ***switch 2***.

The following output displays the three policies as configured on the DMF controller:

```
Controller-1(config)# show policy
# Policy Name                          Action  Runtime Status Type       Priority Overlap Priority Push VLAN Filter BW Delivery BW Post Match Filter Traffic Delivery Traffic Services
-|-----------------------------------|-------|--------------|----------|--------|----------------|---------|---------|-----------|-----------------------|----------------|--------|
1 MLAG-policy                         forward installed      Configured 100      0                3         25Gbps    80Gbps      1.22Gbps                1.23Gbps
2 _mlag_Policy-1_DeliverySwitch-1     forward installed      Dynamic    100      1                2         40Gbps    1Gbps       612Mbps                 612Mbps
3 _mlag_Policy-1_DeliverySwitch-2     forward installed      Dynamic    100      1                1         40Gbps    1Gbps       613Mbps                 613Mbps
Controller-1(config)#
```

Below are the details for each policy:

Policy: *Policy-1* Interfaces

- Filter Interface(s) section lists the filter interface configured for the policy, ***Policy-1***.
- Core Interface(s) section lists the interfaces that connect the filter switch and the core switch selected for the policy.
- MLAG Core Interface(s) section displays the core LAG interface that hashes the traffic towards the peer switches.
- MLAG Delivery Interface(s) section lists the delivery MLAG interface members.

Policy: _mlag_Policy-1_DeliverySwitch-1 Interfaces

- Filter Interfaces(s) section lists the dynamically configured interface name on ***DeliverySwitch1*** to which the core switch is connected.
- MLAG Delivery Interface(s) section lists the delivery MLAG interface member on ***DeliverySwitch1***.

Policy: _mlag_Policy-1_DeliverySwitch-2 Interfaces

- Filter Interfaces(s) section lists the dynamically configured interface name on ***DeliverySwitch2*** to which the core switch is connected.
- MLAG Delivery Interface(s) section lists the delivery MLAG interface member on ***DeliverySwitch2***.

7.9.10 MLAG Link Discovery

Link Layer Discovery Protocol (LLDP) is used to discover MLAG links. When the DMF Controller receives an LLDP message, it looks for the switch and interface names. If the switch is a part of an MLAG domain, and the reported interface corresponds to the MLAG interface, then it is classified as an MLAG link.

```
Controller-1(config)# show link all link-type mlag-member
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Links ~~~~~~~~~~~~~~~~~~~
```

7.9.11 Overlapping Policies in LAGs

An overlapping policy is dynamically configured if two configured policies share at least one filter interface and at least one of the delivery interfaces is different.

When two DMF policies are configured to use an MLAG interface as a delivery interface overlap, the following policies are created:

1. ***Policy-1*** uses the filter interface ***Filter-1*** and the delivery interface ***MLAG-Tool-1***.
2. ***Policy-2*** uses the filter interface ***Filter-1*** and the delivery interface ***MLAG-Tool-2***.
3. The above two policies will result in an overlapping policy. An overlapping policy will be configured following the naming convention ***_Policy-1_o_Policy-2***.

After calculating the overlapping policy for the two user-configured policies, DMF configures two dynamic policies: one for each peer switch in the MLAG domain and one for each of the three policies listed above.

Table 5: Dynamic Policies

MLAG Dynamic Policy	Parent Policy	Delivery Switch/Peer switch
_mlag_Policy-1_DeliverySwitch-1	***Policy-1***	***DeliverySwitch-1***
_mlag_Policy-1_DeliverySwitch-2	***Policy-1***	***DeliverySwitch-2***
_mlag_Policy-2_DeliverySwitch-1	***Policy-2***	***DeliverySwitch-1***
_mlag_Policy-2_DeliverySwitch-2	***Policy-2***	***DeliverySwitch-2***
_mlag Policy-1_o_Policy-2_DeliverySwitch-1	***_Policy-1_o_Policy-2***	***DeliverySwitch-1***
_mlag Policy-1_o_Policy-2_DeliverySwitch-2	***_Policy-1_o_Policy-2***	***DeliverySwitch-2***

The following policies, ***Policy-1*** and ***Policy-2***, share the same filter interface, ***Filter-1***, but are configured to use different delivery interfaces, ***MLAG-Tool-1*** and ***MLAG-Tool-2***. No priority is configured; these policies use the same default priority.

Policy-1 Configuration

```
policy Policy-1
action forward
delivery-interface MLAG-Tool-1
filter-interface Filter-1
1 match ip src-ip 200.200.0.0 255.255.255.0
```

Policy-2 Configuration

```
policy Policy-2
action forward
delivery-interface MLAG-Tool-2
filter-interface Filter-1
1 match ip dst-ip 100.100.0.0 255.255.255.0
```

The above two policies will result in an overlapping policy.

```
Controller-1(config)# show policy
# Policy Name            Action  Runtime Status Type       Priority Overlap Priy Push VLAN Filter BW Delivery BW Post Match Filt Traff Del Traffic Services
-|---------------------|-------|--------------|----------|--------|------------|---------|---------|-----------|---------------------|----------|--------|
1 Policy-1               forward installed      Configured      100            0         1    25Gbps      80Gbps                314Mbps    315Mbps
2 Policy-2               forward installed      Configured      100            0         3    25Gbps      80Gbps                314Mbps    315Mbps
3 _Policy-1_o_Policy-2   forward installed      Dynamic         100            1         5    25Gbps      80Gbps                314Mbps    315Mbps
```

- ***Policy-1***: User-configured policy to forward packets matching source IP ***200.200.0.0/24*** to ***MLAG-Tool-1***.
- ***Policy-2***: User-configured policy to forward packets matching destination IP ***100.100.0.0/24*** to ***MLAG-Tool-2***.
- ***_Policy-1_o_Policy-2***: A dynamically configured overlapping policy with higher Overlap Priority to ensure that if a packet matches rules from both the policies (source IP of ***200.200.0.1*** and destination IP of ***100.100.0.1***), it forwards to both ***MLAG-Tool-1*** and ***MLAG-Tool-2***.

The following are the dynamic policies configured for each delivery switch in the MLAG Domain.

Policy-1 Dynamic Policies

```
# Policy Name                                Action  Runtime Status Type       Priority Overlap Priority Push VLAN Filter BW Delivery BW Post Match Fil Traff Del Traff Services
-|-----------------------------------------|-------|--------------|----------|--------|----------------|---------|---------|-----------|--------------------|---------|--------|
4 _mlag_Policy-1_DeliverySwitch-1           forward installed      Dynamic         100              1         2    40Gbps      1Gbps                       -
5 _mlag_Policy-1_DeliverySwitch-2           forward installed      Dynamic         100              1         6    40Gbps      1Gbps
```

- ***_mlag_Policy-1_DeliverySwitch-1***: MLAG dynamic policy for ***Policy-1*** for ***DeliverySwitch-1***.
- ***_mlag_Policy-1_DeliverySwitch-2***: MLAG dynamic policy for ***Policy-1*** for ***DeliverySwitch-2***.

Policy-2 Dynamic Policies

```
# Policy Name                          Action Runtime Status Type       Priority Overlap Pri Push VLAN Filter BW Delivery BW Post Match Fil Traff Del Traff Services
-|-----------------------------------|-------|--------------|----------|--------|-----------|---------|---------|-----------|--------------------|---------|--------|
6 _mlag_Policy-2_DeliverySwitch-1    forward installed      Dynamic         100              1         9    40Gbps      10Gbps                      -
7 _mlag_Policy-2_DeliverySwitch-2    forward installed      Dynamic         100              1         7    40Gbps      10Gbps                      -
```

- ***_mlag_Policy-2_DeliverySwitch-1***: MLAG dynamic policy for ***Policy-2*** for ***DeliverySwitch-1***.
- ***_mlag_Policy-2_DeliverySwitch-2***: MLAG dynamic policy for ***Policy-2*** for ***DeliverySwitch-2***.

_Policy-1_o_Policy-2 Dynamic Policies

The following policies have higher Overlap Priority than the rest to prioritize the overlapping traffic forwarded to ***DeliverySwitch-1*** and ***DeliverySwitch-2***.

```
# Policy Name                                  Action  Runtime Status Type       Priority Overlap Priority Push VLAN Filter BW Delivery BW Post Match Filter Traffic Delivery Traffic Services
-|-------------------------------------------|-------|--------------|----------|--------|----------------|---------|---------|-----------|---------------------|----------------|--------|
8 _mlag__Policy-1_o_Policy-2_DeliverySwitch-1  forward installed      Dynamic         100              3         4    40Gbps      11Gbps                157Mbps          315Mbps
9 _mlag__Policy-1_o_Policy-2_DeliverySwitch-2  forward installed      Dynamic         100              3         8    40Gbps      11Gbps                157Mbps          315Mbps
```

- ***_mlag Policy-1_o_Policy-2_DeliverySwitch-1***: MLAG dynamic policy for overlapping policy for ***DeliverySwitch-1***.
- ***_mlag Policy-1_o_Policy-2_DeliverySwitch-2***: MLAG dynamic policy for overlapping policy for ***DeliverySwitch-2***.

7.9.12 Using LAG Interfaces as Members in MLAG Interfaces

MLAG interface members can be physical interfaces or LAG interfaces to increase bandwidth. To add a LAG member to an MLAG interface, use the following procedure:

1. Configure the LAG interface on Peer Switch 1.

```
Controller-1(config)# switch DeliverySwitch-1
Controller-1(config-switch)# lag-interface LAG-peer-switch-1
Controller-1(config-switch-lag-if)# member ethernet11
Controller-1(config-switch-lag-if)# member ethernet12
```

2. Configure the LAG interface on Peer Switch 2.

```
Controller-1(config)# switch DeliverySwitch-2
Controller-1(config-switch)# lag-interface LAG-peer-switch-2
Controller-1(config-switch-lag-if)# member ethernet11
Controller-1(config-switch-lag-if)# member ethernet12
```

3. Add the configured LAG interfaces as members into the MLAG interface.

```
Controller-1(config)# mlag-domain Domain1
Controller-1(config-mlag-domain)# mlag-interface MLAG-LAG-Del-Intf
Controller-1(config-mlag-domain-if)# member switch DeliverySwitch-1 interface LAG-peer-switch-1
Controller-1(config-mlag-domain-if)# member switch DeliverySwitch-2 interface LAG-peer-switch-2
Controller-1(config-mlag-domain-if)# role delivery interface-name MLAG-LAG-Tool-1
```

4. Configure the DMF policy using the delivery interface `MLAG-LAG-Tool-1`.

```
Controller-1(config)# policy Policy-1
Controller-1(config-policy)# action forward
Controller-1(config-policy)# 1 match any
Controller-1(config-policy)# filter-interface Filter-1
Controller-1(config-policy)# delivery-interface MLAG-LAG-Tool-1
```



Note: Traffic will not hash toward tools if the core switch LAG has the same number of member interfaces as the LAG on the peer switches of MLAG delivery.

Workaround:

- Ensure that the hash fields on the core switch and MLAG peer switches are different or:
- Ensure that the number of member interfaces in the LAG interface configured on the core switch differs from the number of members in the LAG interface configured on peer switches.

Chapter 8

Tunneling Between Data Centers

This chapter describes establishing Generic Routing Encapsulation (GRE) and or Virtual Extensible LAN (VXLAN) tunnels between DMF switches in different locations or between a DMF switch and a third-party device.

8.1 Understanding Tunneling

DMF can forward traffic between two DMF switches controlled by the same Controller over a tunnel. Use this feature to extend a DMF deployment across multiple data centers or branch offices over networks connected by Layer-3 networks. This feature supports the centralization or distribution of tools and taps across multiple locations when they cannot be cabled directly.

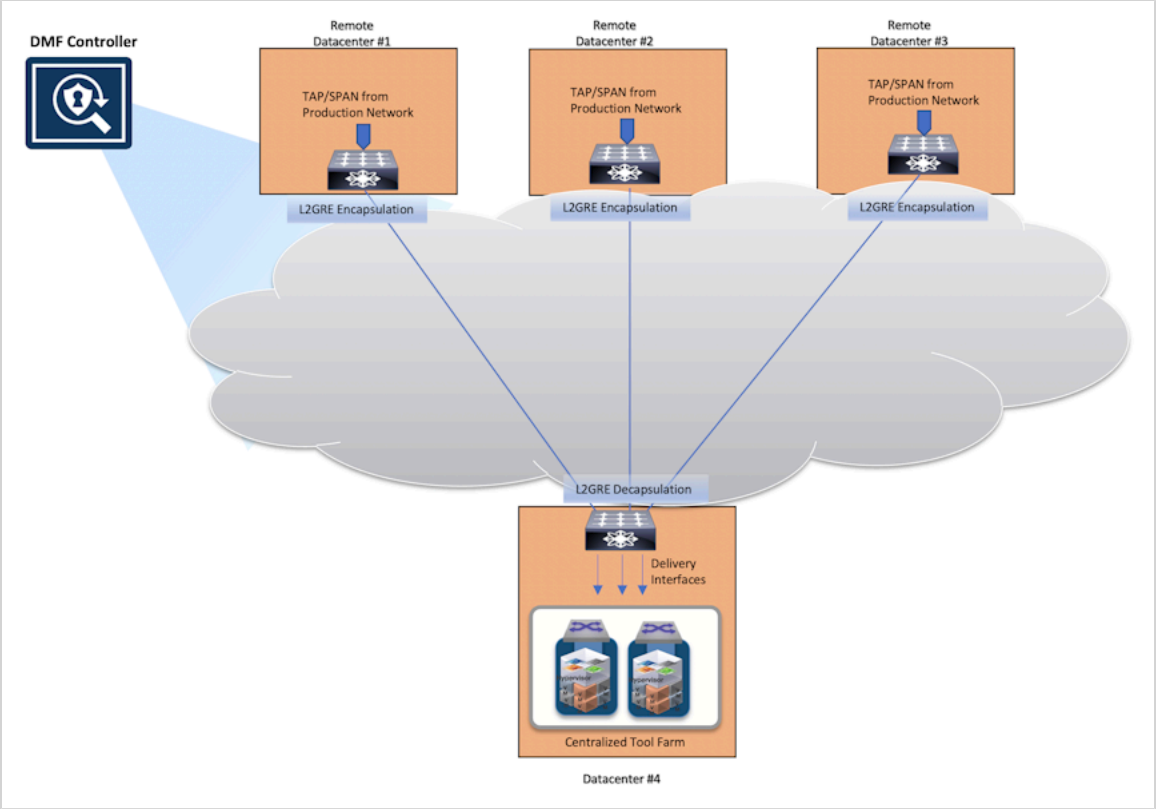


Note: Refer to the ***DANZ Monitoring Fabric 8.6 Hardware Compatibility List*** for a list of the switches that support tunneling. The ***DANZ Monitoring Fabric 8.6 Verified Scale Guide*** indicates the number of tunnels supported by each supported switch (Verified Scalability Values/Encap Tunnels/Decap Tunnels).

When enabling tunneling between DMF switches, keep the following in mind:

- Connect switch ports in the main data center and the remote location to the appropriate WAN routers and ping each interface to ensure IP connectivity is established.
- Create tunnel endpoints and configure the tunnel attributes on each end of the tunnel.
- The **CRC Check** option must be enabled if tunneling is enabled, which it is by default. If CRC checking is disabled, re-enable it before configuring a tunnel.
- In the case of GRE tunnels, the optional **gre-key-decap** value on the receiving end must match the GRE key value of the sender. The option exists to set multiple values on the same tunnel to decapsulate traffic with different keys.
- A single switch can initiate multiple tunnels. Configure a separate encap-loopback-interface for each tunnel (transmit-only or bidirectional).
- Set the **loopback-mode** to ***mac*** on the ***encap-loopback-interface***.

Figure 8-1: Connecting DMF Switches Using a Layer-2 GRE Tunnel



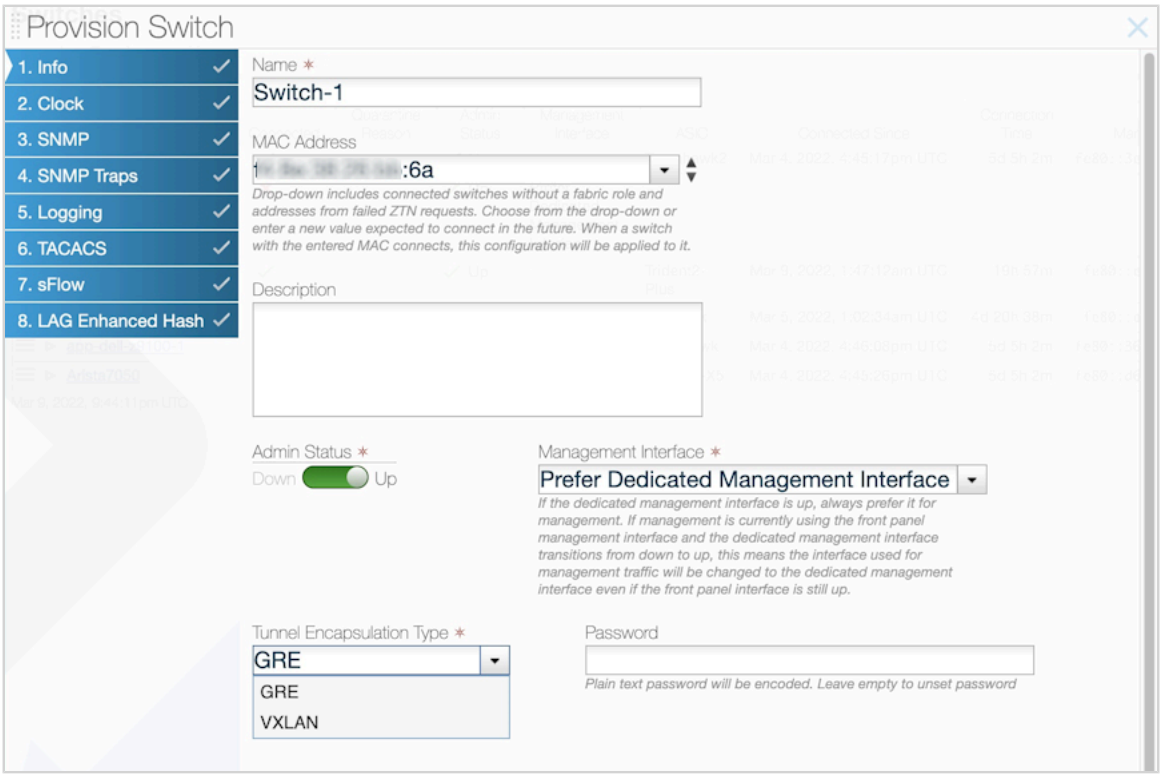
Note: For EOS switches running DMF 8.5: L2GRE tunneling is supported on Arista 7280R3 switches only and subject to the following limitations:

- L2GRE tunnels are not supported on DMF 7280R and 7280R2 switches.
- DSCP configuration is not supported.
- Traffic steering for traffic arriving on an L2GRE tunnel will only allow for matching based on inner src/dst IP, IP protocol, and inner L4 src/dst port.
- Packets may only be redirected to a single L2GRE tunnel.
- Packets may not be load-balanced across multiple L2GRE tunnels.
- Only IPv4 underlays in the default VRF are supported.
- Matching on inner IPv6 headers may not be supported.
- The maximum number of tunnels on EOS Jericho switches is 32.
- There is no bi-directional tunnel support. The parent/uplink router-facing interface is used for either encapsulation or decapsulation, but not simultaneously.
- When using tunnel-as-a-filter, there is no inner L3/L4 matching support immediately after decapsulation in the same switch pass. Using a loopback may work around this limitation.
- VXLAN tunnels are currently NOT supported on 7280 switches.

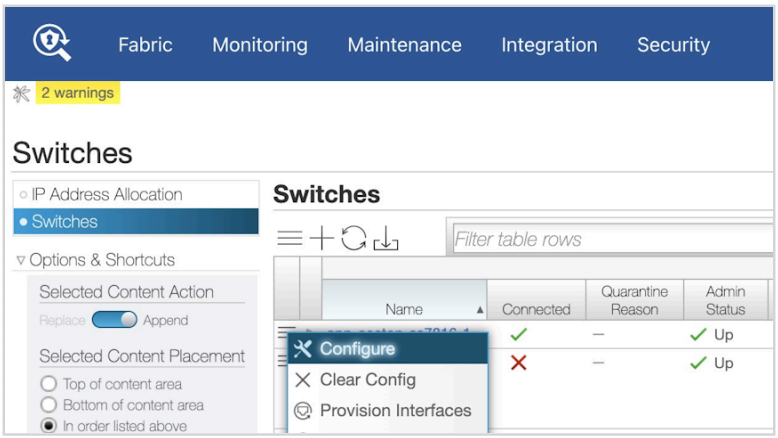
8.2 Encapsulation Type

DANZ Monitoring Fabric (DMF) supports VXLAN tunnel type and Level-2 Generic Routing Encapsulation (L2GRE). The tunnel type is a per-switch configuration, setting the switch pipeline to VXLAN or L2GRE. Once the switch pipeline is set, all tunnels configured on the switch will use the same tunnel type.

The encapsulation type can be configured in the GUI while adding a new switch into the DMF Controller, as shown in the figure below:



The encapsulation type can be edited for an existing switch from the **Fabric > Switches > Configure Switch** page, as shown in the figure below:



8.3 Using Tunnels in Policies

Tunnels can be used as a core link, filter interface, or delivery interface. The most common use case is linking multiple sites, using the tunnel as a core link. If used as a core link, DMF automatically discovers the link as if it were a physical link and similarly determines connectivity (link-state). If the tunnel goes down for any reason, DMF treats the failure as it would a physical link failure.

Another typical use case for the tunnel is as a filter interface to decapsulate L2 GRE/VXLAN tunneled production traffic or a tunnel initiated by another DMF instance managed by a different DMF Controller. Use the tunnel endpoint as a delivery interface to encapsulate filtered monitoring traffic to send to analysis tools or another DANZ Monitoring Fabric managed by a different DMF Controller.



Note: By default, sFlow®* and other Arista Analytics metadata cannot be generated for decapsulated L2 GRE/VXLAN tunneled production traffic on a tunnel interface configured as a filter interface. To generate this metadata, create a policy with a filter interface as a tunnel interface and send the decapsulated traffic to a MAC loopback port configured in a filter-and-delivery role. Now, create a second policy with the filter interface as the MAC loopback port and the delivery interface going to the tools. The sFlow and metadata will now be generated for the decapsulated tunnel traffic.

* sFlow® is a registered trademark of Inmon Corp.

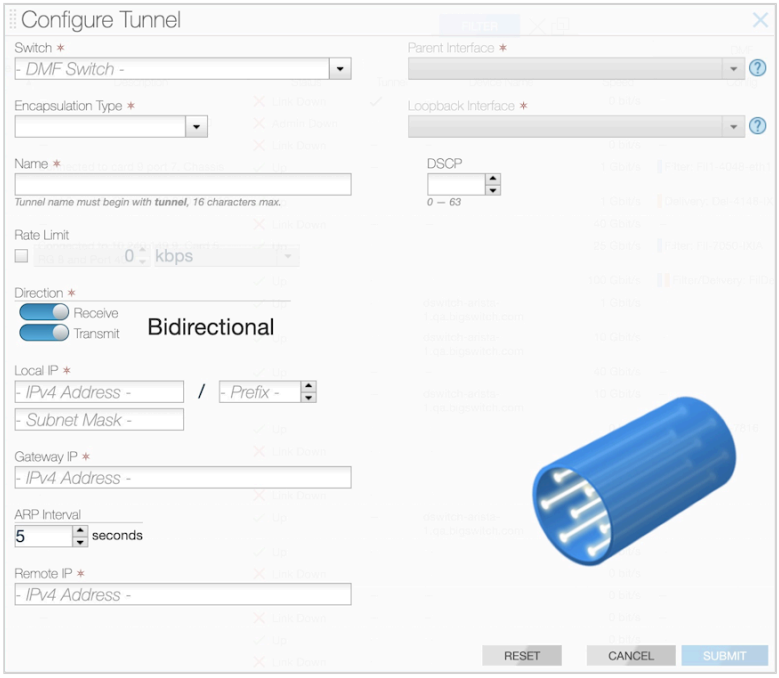
8.4 Configure a GRE Tunnel

To configure a VXLAN tunnel, perform the following steps:

1. Select **Fabric > Switch**.
2. On the **Switches** page, select the **Menu** control next to the switch or interface to include in the tunnel and select **Create Tunnel**.

   Alternatively, configure tunnels from the **Fabric > Interfaces** page by selecting the **Menu Control > Create Tunnel** option. The system displays the dialog as shown in the figure below:

   Figure 8-2: Configure VXLAN Tunnel



3. Complete the fields on this page as described below.
   - **Switch**: From the drop-down, select the DMF switch.
   - **Encapsulation Type**: The encapsulation type will automatically be selected based on the pipeline mode of the selected switch.
   - **Name**: Name of the tunnel, beginning with the word `tunnel`.
   - **Rate Limit** (Optional): Packets entering the tunnel can be rate-limited to restrict the bandwidth usage of the tunnel. This can help ensure that a WAN link is not saturated with monitoring traffic being tunneled between sites. This setting is applicable on the tunnel encapsulation side.
   - **Direction**: Direction can be bidirectional, transmit-only, or receive-only. For bidirectional tunnels, set the tunnel direction to bidirectional on both ends. For uni-directional tunnels from remote to main datacenter, the tunnel direction is transmit-only on the remote datacenter switch and receive-only on the main data center switch.
   - **Local IP**: Local IP address and subnet mask in CIDR format (/nn).

- **Gateway IP**: IP address of the default (next-hop) gateway.
- **Remote IP**: This is the IPv4 address of the other end (remote end) of the tunnel.
- **Parent Interface**: Physical port or port-channel interface associated with the tunnel. This is the destination interface for the tunnel.
- **Loopback Interface**: A physical interface on each switch with a transmit-only or a bidirectional tunnel endpoint. Use this physical interface for tunnel encapsulation and not for any other DMF purpose, such as a filter, delivery, service, or core interface.
- **DSCP** (Optional): Mark the tunnel traffic with the specified DSCP value.

4. After configuring the appropriate options, select **Submit**.



Note: Configure this procedure on both switches at each end of the tunnel. Set the **Auto VLAN** mode to **`Push Per Policy`** or **`Push Per Filter Interface`**.

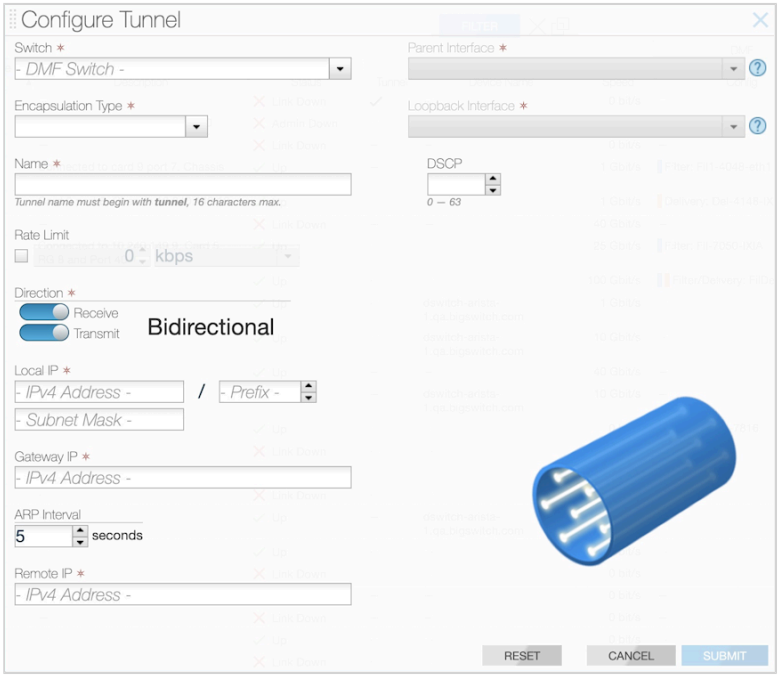
8.5 Configure a VXLAN Tunnel

To configure a VXLAN tunnel using the GUI, perform the following steps:

1. Select **Fabric > Switch**.
2. On the **Switches** page, select the **Menu** control next to the switch or interface to include in the tunnel and select **Create Tunnel**.

   Alternatively, configure tunnels from the **Fabric > Interfaces** page by selecting the **Menu Control > Create Tunnel** option. The system displays the dialog as shown in the figure below:

   Figure 8-3: Configure VXLAN Tunnel



3. Complete the fields on this page as described below:
   - **Switch**: From the drop-down, select the DMF switch.
   - **Encapsulation Type**: Encapsulation type will automatically be selected based on the pipeline mode of the selected switch.
   - **Name**: Name of the tunnel, beginning with the word **`tunnel`**.
   - **Rate Limit** (Optional): Packets entering the tunnel can be rate-limited to restrict bandwidth usage of the tunnel. This can help ensure that a WAN link is not saturated with monitoring traffic being tunneled between sites. This setting is applicable on the tunnel encap side.
   - **Direction**: bidirectional, transmit-only, or receive-only. For bidirectional tunnels, set tunnel-direction to bidirectional on both ends. For uni-directional tunnels from remote to main datacenter, tunnel-direction is transmit only on the remote datacenter switch and the tunnel-direction is receive-only on the main data center switch.
   - **Local IP**: Local IP address and subnet mask in CIDR format (/nn).

- **Gateway IP**: IP address of the default (next-hop) gateway.
- **Remote IP**: This is the IPv4 address of the other end (remote end) of the tunnel.
- **Parent Interface**: Physical port or port-channel interface associated with the tunnel. This is the destination interface for the tunnel.
- **Loopback Interface**: A physical interface on each switch with a transmit-only or a bidirectional tunnel endpoint. Use this physical interface for tunnel encapsulation and not for any other DMF purpose, such as a filter, delivery, service, or core interface.
- **DSCP** (Optional): Mark the tunnel traffic with the specified DSCP value.

4. After configuring the appropriate options, select **Submit**.



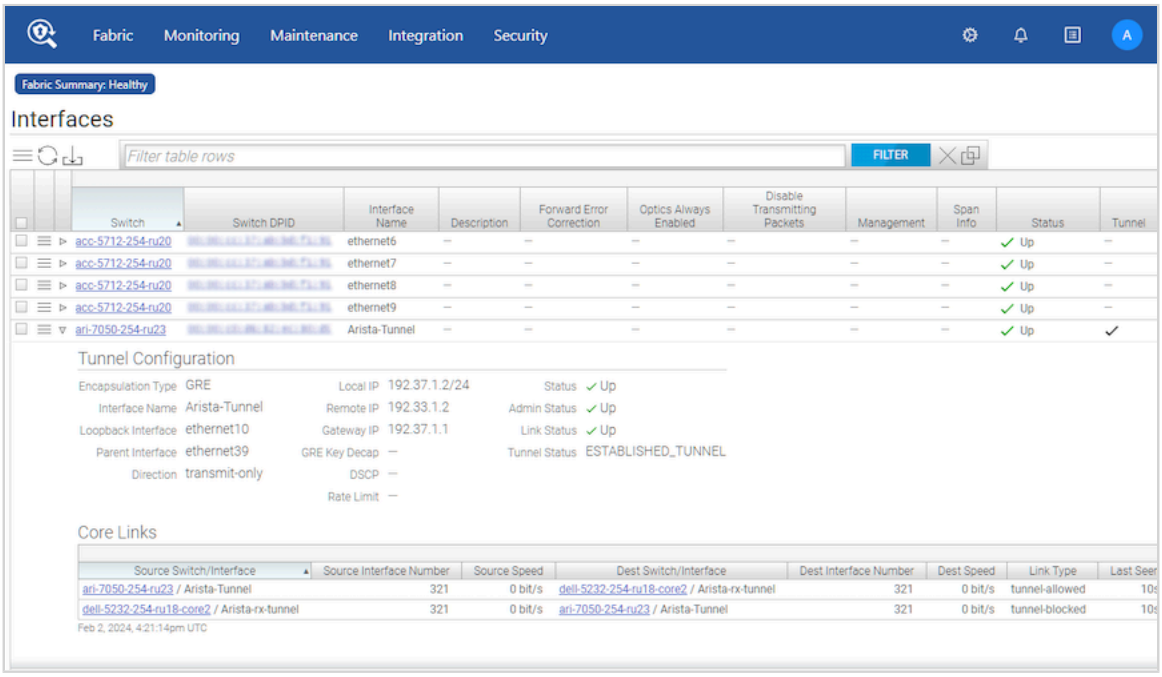
Note: Configure this procedure on both switches at each end of the tunnel. Set the **Auto VLAN** mode to **`Push Per Policy`** or **`Push Per Filter Interface`**.

8.6 Viewing or Modifying Existing Tunnels

To view or modify the configuration of an existing tunnel, use the **Fabric > Interfaces** option. To view the tunnel configuration, expand the interface. DMF displays the tunnel configuration as illustrated in the following figure.

> **Tip:** When multiple interfaces are present, use the **Filter** feature to locate tunnel interfaces after typing the first few letters of the word *tunnel*.

Figure 8-4: Tunnel Interfaces



The expanded row displays the status and other properties of the tunnel configured for the selected interface. Use the **Menu** control and select **Configure Tunnel** to modify the tunnel configuration. Select **Delete Tunnel** to remove the tunnel.

8.7 Using a Tunnel with User-defined Offsets

With an L2-GRE or VXLAN tunnel, matching traffic on a user-defined offset results in dropping interesting traffic. The tunnel header throws off the offset calculation, and the selected traffic may be dropped. This behavior is due to how switch hardware calculates the anchor and offset concerning incoming packets. When the core link is a tunnel, the anchor and offset calculation differs when encapsulating packets compared to when decapsulating.

There are two ways to work around this issue:

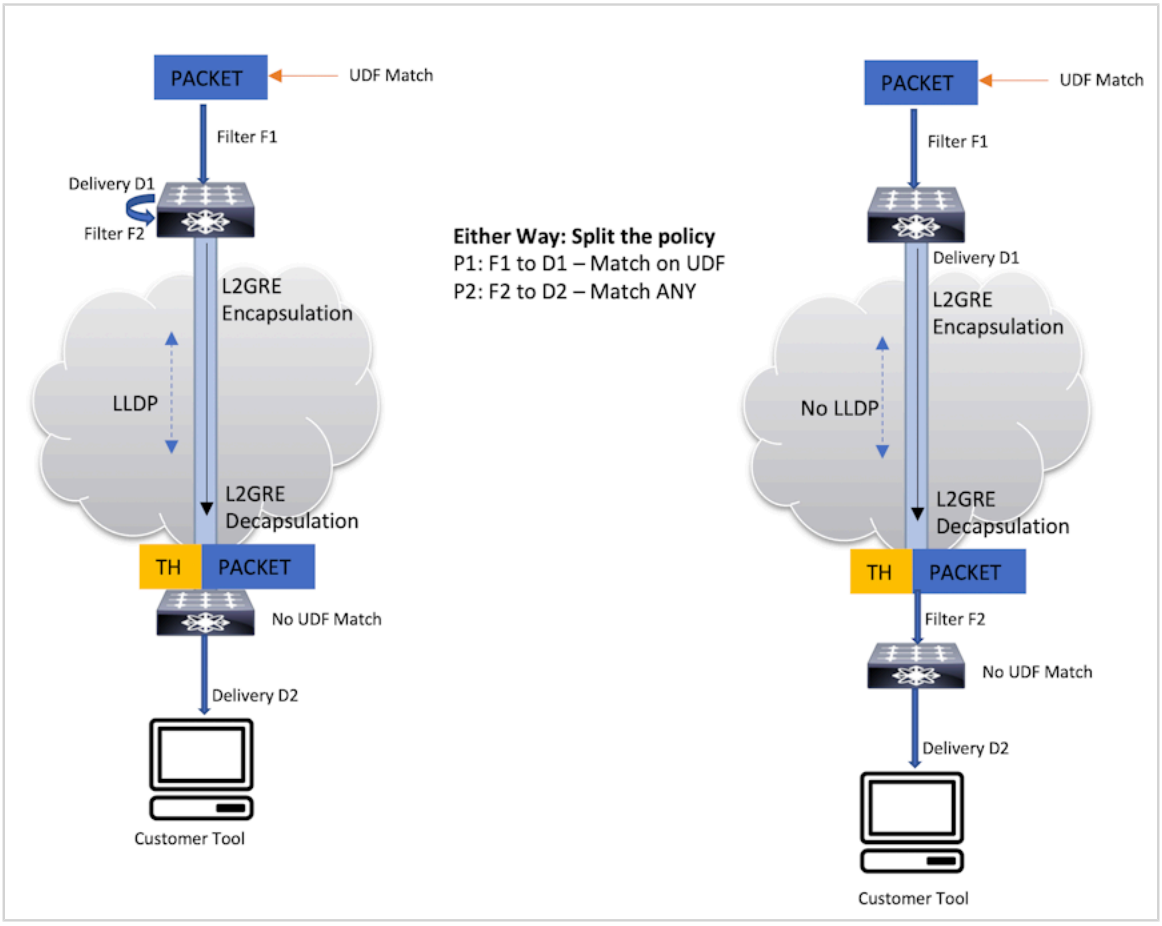
- Avoid matching on user-defined offsets on tunnel interfaces
- Avoid using a tunnel as a core link when matching on a user-defined offset

Avoid matching on user-defined offsets when the ingress filter interface is a tunnel by filtering on the user-defined offset before the traffic enters a tunnel used as a filter interface. This preserves the LLDP messaging on the core tunnel link, but it requires an extra physical loopback interface on the encapsulating switch. The figure below illustrates both of these workarounds. In either case, a UDF match is applied to the ingress traffic on filter interface F. For example, the policy might apply a match at offset 20 after the start of the L4 header. In both workarounds, the policy has been split into two policies:

P1: ***F*** to ***D1***, match on user-defined offset ***P2***: ***F1*** to ***D***, match any.

In the example on the left, the ingress interface on the decapsulating switch, which is included in a core tunnel link, no longer has to calculate the user-defined offset. This solution preserves LLDP messages on the tunnel link but requires an extra loopback interface.

Figure 8-5: Using Tunnels with User-Defined Offsets



In the example on the right, the tunnel endpoints are configured as filter and delivery interfaces. This solution avoids using the tunnel as a core link and does not require an extra physical loopback interface. However, LLDP updates are lost on the tunnel link.

8.8 Configuring Wildcard Tunnels

Wildcard Tunnels on SAND Platforms

The Wildcard tunneling feature allows the DANZ Monitoring Fabric (DMF) to decapsulate L2GRE-based tunneled traffic from any remote source. This feature, supported on Switch Light OS (SWL) based DMF switches in prior releases, now allows wildcard tunnels on Arista EOS-based DMF switches.

Platform Compatibility

EOS switches with Jericho2 or higher ASICs compatible with DMF 8.5.0 that support L2GRE tunneling. L2GRE tunneling on EOS SAND platforms is only supported on Arista 7280R3 switches.

Enable **Tunneling** by navigating to the **DMF Features** page and selecting the **gear** icon in the navigation bar.

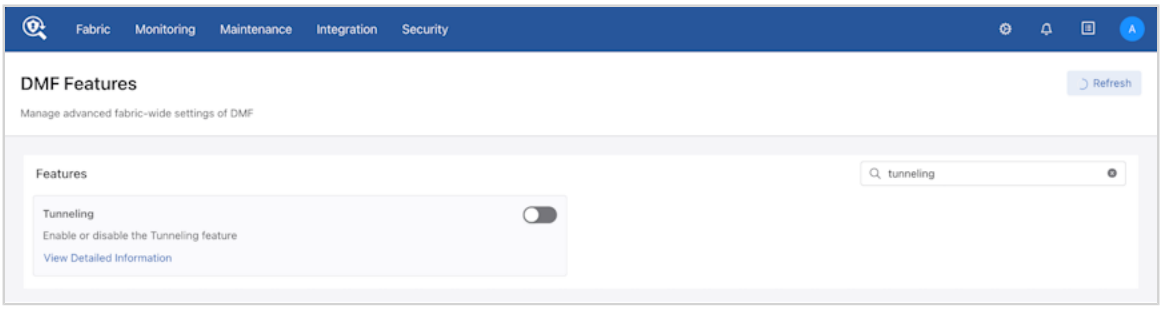
Figure 8-6: DMF Navigation Menu



Locate the **Tunneling** card feature on the page.

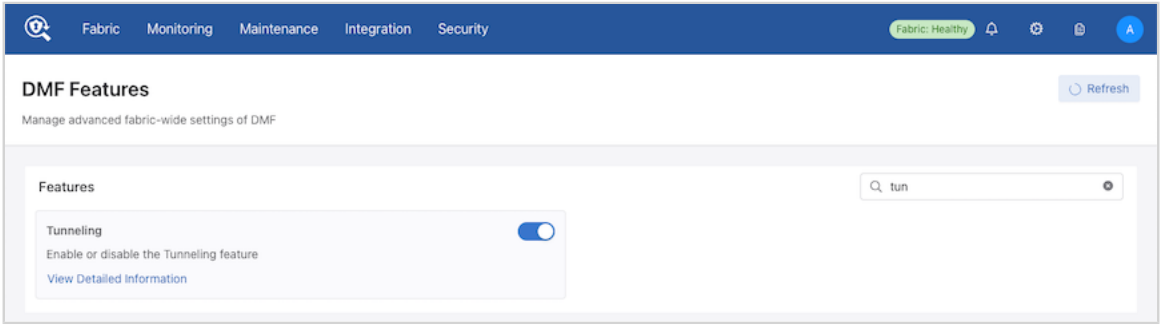
> **Tip:** Use the search bar to quickly locate the feature.

Figure 8-7: DMF Features - Tunneling



Select the **toggle** switch to enable the Tunneling feature on DMF.

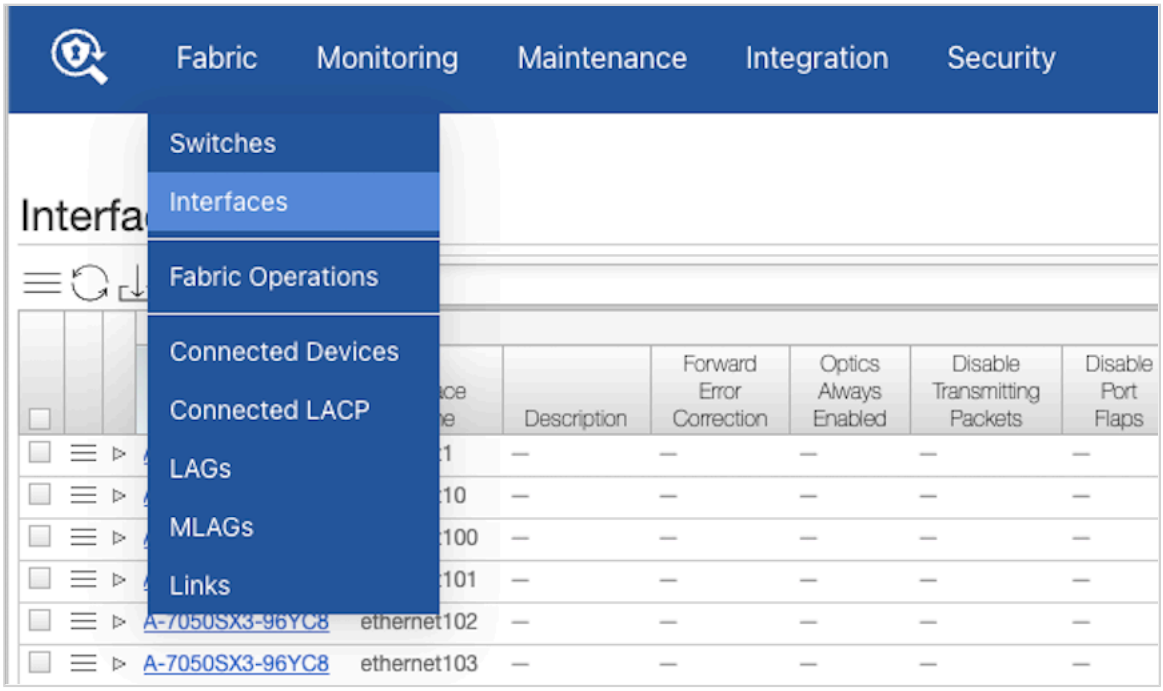
Figure 8-8: Tunneling On



Steps to Enable Wildcard Tunnels

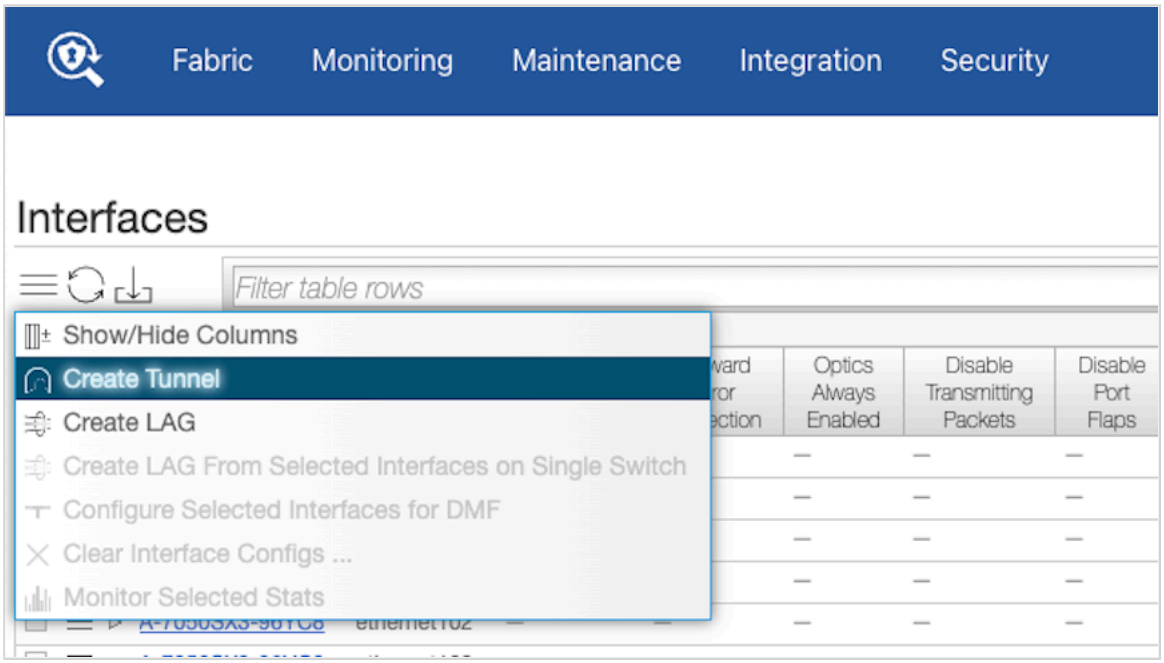
1. Locate the **Switches** tab on the **Fabric > Switches** page.

   Figure 8-9: Fabric > Interfaces



2. Select the **Create Tunnel** option from the table menu.

Figure 8-10: Create Tunnel



3. Create a **Tunnel** (select **Encapsulation Type** as **GRE**) by entering the required fields. To enable wildcard support, enter the **Remote IP** input with data as shown below.

Figure 8-11: Configure Tunnel



4. Select **Submit** to save the tunnel configuration.

8.9 Using the Command Line Interface

Configure a GRE Tunnel

Configure a VXLAN Tunnel

Configuring Wildcard Tunnels

Encapsulation Types

8.9.1 Configure a GRE Tunnel

To configure a GRE tunnel using the CLI, perform the following steps:

1. Connect switch ports (on remote and main datacenter) to their respective WAN routers and ensure they can communicate via IP.
2. Enable tunneling on the DMF network by entering the following command from config mode:

```
controller-1(config)# tunneling
Tunneling is an Arista Licensed feature. Please ensure that you have purchased the license
for tunneling before using this feature. enter "yes" (or "y") to continue: yes
controller-1(config)#
```

3. Configure the MAC loopback mode, as shown in the following example:

```
controller-1(config)# switch DMF-CORE-SWITCH
controller-1(config-switch)# interface ethernet7
controller-1(config-switch-if)# loopback-mode mac
```

4. Create tunnel endpoints.

The following CLI example configures a bi-directional tunnel from ***remote-dc1-filter-sw*** to ***main-dc-delivery-sw***:

```
!
switch remote-dc1-filter-sw
gre-tunnel-interface tunnel1
remote-ip 192.168.200.50
gre-key-decap 4097 === 4097 is the VPN key used for the tunnel ID
parent-interface ethernet6
local-ip 192.168.100.50 mask 255.255.255.0 gateway-ip 192.168.100.1
direction bidirectional encap-loopback-interface ethernet38
!
switch main-dc-delivery-sw
gre-tunnel-interface tunnel1
remote-ip 192.168.100.50
gre-key-decap 4097 === 4097 is the VPN key used for the tunnel ID
parent-interface ethernet5
local-ip 192.168.200.50 mask 255.255.255.0 gateway-ip 192.168.200.1
direction bidirectional encap-loopback-interface ethernet3
```

The following CLI example configures a uni-directional tunnel from ***remote-dc1-filter-sw*** to ***main-dc-delivery-sw***:

```
!
switch remote-dc1-filter-sw
gre-tunnel-interface tunnel1
remote-ip 192.168.200.50
gre-key-decap 4097 === 4097 is the VPN key used for the tunnel ID
interface parent-interface ethernet6
local-ip 192.168.100.50 mask 255.255.255.0 gateway-ip 192.168.100.1
direction transmit-only encap-loopback-interface ethernet38
!
switch main-dc-delivery-sw
gre-tunnel-interface tunnel1
remote-ip 192.168.100.50
gre-key-decap 4097 === 4097 is the VPN key used for the tunnel ID
parent-interface ethernet5
local-ip 192.168.200.50 mask 255.255.255.0 gateway-ip 192.168.200.1
direction receive-only
```

8.9.1.1 Rate Limit the Packets on a GRE Tunnel

Packets entering the GRE tunnel can be rate-limited to limit bandwidth usage by the tunnel and help ensure that a WAN link is not saturated with monitoring traffic being tunneled between sites. This setting is applicable on the tunnel encapsulation side.



Note: The minimum recommended value for rate limiting on the tunnel interface is ***25*** kbps. When attempting to set a value below this limit, the switch will still set the rate limit value to ***25*** kbps.

```
switch DMF-CORE-SWITCH-1
gre-tunnel-interface tunnel1
direction bidirectional encap-loopback-interface ethernet10
------------------------------example truncated------------
interface ethernet10
rate-limit 1000
```

8.9.1.2 View GRE Tunnel Interfaces

All CLI **show** commands for regular interfaces apply to GRE tunnel interfaces.

Use the **show running-config** command to view the configuration of tunnel interfaces.

Enter the **show tunnel** command to see a tunnel interface's configuration parameters and runtime state.

```
controller-1# show tunnel
# Switch DPID               Tunnel Name Tunnel Status     Direction     Src IP       Dst IP       Parent Name  Loopback Name
-|------------------------|-----------|-----------------|-------------|------------|------------|------------|-------------|
1 DMF-CORE-SWITCH-1         tunnel1     ESTABLISHED_TUNNEL bidirectional 198.82.215.1 216.47.143.1 ethernet5:1 ethernet6
2 DMF-CORE-SWITCH-2         tunnel1     ESTABLISHED_TUNNEL bidirectional 216.47.143.1 198.82.215.1 ethernet11:3 ethernet5
3 DMF-CORE-SWITCH-2         tunnel2     ESTABLISHED_TUNNEL bidirectional 192.168.43.1 192.168.42.1 ethernet11:4 ethernet17
4 DMF-CORE-SWITCH-3         tunnel2     ESTABLISHED_TUNNEL bidirectional 192.168.42.1 192.168.43.1 ethernet6 ethernet33

controller-1# show tunnel switch DMF-CORE-SWITCH-2
# Switch DPID               Tunnel Name Tunnel Status     Direction     Src IP       Dst IP       Parent Name  Loopback Name
-|------------------------|-----------|-----------------|-------------|------------|------------|------------|-------------|
1 DMF-CORE-SWITCH-2         tunnel1     ESTABLISHED_TUNNEL bidirectional 216.47.143.1 198.82.215.1 ethernet11:3 ethernet5
2 DMF-CORE-SWITCH-2         tunnel2     ESTABLISHED_TUNNEL bidirectional 192.168.43.1 192.168.42.1 ethernet11:4 ethernet17

controller-1# show tunnel switch DMF-CORE-SWITCH-2 tunnel1
# Switch DPID               Tunnel Name Tunnel Status     Direction     Src IP       Dst IP       Parent Name  Loopback Name
-|------------------------|-----------|-----------------|-------------|------------|------------|------------|-------------|
1 DMF-CORE-SWITCH-2         tunnel1     ESTABLISHED_TUNNEL bidirectional 216.47.143.1 198.82.215.1 ethernet11:3 ethernet5
controller-1#
```

8.9.2 Configure a VXLAN Tunnel

To configure a VXLAN tunnel using the CLI, perform the following steps:

1. Connect switch ports (on remote and main datacenter) to their respective WAN routers and ensure that they can communicate via IP.
2. Enable tunneling on the DMF network by entering the following command from config mode:

```
controller-1(config)# tunneling
Tunneling is an Arista Licensed feature. Please ensure that you have purchased the license
for tunneling before using this feature. enter "yes" (or "y") to continue: yes
controller-1(config)#
```

3. Configure the MAC loopback mode, as shown in the following example:

```
controller-1(config)# switch DMF-CORE-SWITCH
controller-1(config-switch)# interface ethernet7
controller-1(config-switch-if)# loopback-mode mac
```

4. Create tunnel endpoints.

The following CLI example configures a bi-directional tunnel from ***remote-dc1-filter-sw*** to ***main-dc-delivery-sw***:

```
!
switch remote-dc1-filter-sw
vxlan-tunnel-interface tunnel1
remote-ip 192.168.200.50
parent-interface ethernet6
local-ip 192.168.100.50 mask 255.255.255.0 gateway-ip 192.168.100.1
direction bidirectional encap-loopback-interface ethernet38
!
switch main-dc-delivery-sw
vxlan-tunnel-interface tunnel1
remote-ip 192.168.100.50
parent-interface ethernet5
local-ip 192.168.200.50 mask 255.255.255.0 gateway-ip 192.168.200.1
direction bidirectional encap-loopback-interface ethernet3
```

The following CLI example configures a uni-directional tunnel from ***remote-dc1-filter-sw*** to ***main-dc-delivery-sw***:

```
!
switch remote-dc1-filter-sw
vxlan-tunnel-interface tunnel1
remote-ip 192.168.200.50
interface parent-interface ethernet6
local-ip 192.168.100.50 mask 255.255.255.0 gateway-ip 192.168.100.1
direction transmit-only encap-loopback-interface ethernet38
!
switch main-dc-delivery-sw
vxlan-tunnel-interface tunnel1
remote-ip 192.168.100.50
parent-interface ethernet5
local-ip 192.168.200.50 mask 255.255.255.0 gateway-ip 192.168.200.1
direction receive-only
```

8.9.2.1 Rate Limit the Packets on a VXLAN Tunnel

Packets entering the VXLAN tunnel can be rate-limited to limit bandwidth usage by the tunnel and help ensure that a WAN link is not saturated with monitoring traffic being tunneled between sites. This setting is applicable on the tunnel encapsulation side.



Note: The minimum recommended value for rate limiting on the tunnel interface is ***25*** kbps. When attempting to set a value below this limit, the switch will still set the rate limit value to ***25*** kbps.

```
switch DMF-CORE-SWITCH-1
vxlan-tunnel-interface tunnel1
direction bidirectional encap-loopback-interface ethernet10
<snip>
interface ethernet10
rate-limit 1000
```

8.9.2.2 View VXLAN Tunnel Interfaces

All CLI **`show`** commands for regular interfaces apply to tunnel interfaces.

Use the **`show running-config`** command to display the configuration of tunnel interfaces.

Enter the **`show tunnel`** command to see the configuration parameters and runtime state for a VXLAN tunnel interface.

```
controller-1# show tunnel
# Switch DPID Tunnel Name Tunnel Status Direction Src IP Dst IP Parent Name Loopback
Name
-|------------------------|-----------|-----------------|-------------|------------|------------|------------|-------------|
1 DMF-CORE-SWITCH-1 tunnel1 ESTABLISHED_TUNNEL bidirectional 198.82.215.1 216.47.143.1 ethernet5:1 ethernet6
2 DMF-CORE-SWITCH-2 tunnel1 ESTABLISHED_TUNNEL bidirectional 216.47.143.1 198.82.215.1 ethernet11:3 ethernet5
3 DMF-CORE-SWITCH-2 tunnel2 ESTABLISHED_TUNNEL bidirectional 192.168.43.1 192.168.42.1 ethernet11:4 ethernet17
4 dMF-CORE-SWITCH-3 tunnel2 ESTABLISHED_TUNNEL bidirectional 192.168.42.1 192.168.43.1 ethernet6 ethernet33
controller-1#
controller-1# show tunnel switch DMF-CORE-SWITCH-2
# Switch DPID Tunnel Name Tunnel Status Direction Src IP Dst IP Parent Name Loopback Name
-|----------------------|-----------|-----------------|-------------|------------|------------|------------|-------------|
1 DMF-CORE-SWITCH-2 tunnel1 ESTABLISHED_TUNNEL bidirectional 216.47.143.1 198.82.215.1 ethernet11:3 ethernet5
2 DMF-CORE-SWITCH-2 tunnel2 ESTABLISHED_TUNNEL bidirectional 192.168.43.1 192.168.42.1 ethernet11:4 ethernet17
controller-1#
controller-1# show tunnel switch DMF-CORE-SWITCH-2 tunnel1
# Switch DPID Tunnel Name Tunnel Status Direction Src IP Dst IP Parent Name Loopback Name
-|----------------------|-----------|-----------------|-------------|------------|------------|------------|-------------|
1 DMF-CORE-SWITCH-2 tunnel1 ESTABLISHED_TUNNEL bidirectional 216.47.143.1 198.82.215.1 ethernet11:3 ethernet5
controller-1#
```

8.9.3 Configuring Wildcard Tunnels

Wildcard Tunnels on SAND Platforms

The Wildcard tunneling feature allows the DANZ Monitoring Fabric (DMF) to decapsulate L2GRE-based tunneled traffic from any remote source. This feature, supported on Switch Light OS (SWL) based DMF switches in prior releases, now allows wildcard tunnels on Arista EOS-based DMF switches.

Platform Compatibility

EOS switches with Jericho2 or higher ASICs compatible with DMF 8.5.0 that support L2GRE tunneling. L2GRE tunneling on EOS SAND platforms is only supported on Arista 7280R3 switches.

Perform the following steps to configure the tunnels.

1. Enable the tunneling feature before configuring tunnels on a switch using the **`tunneling`** command. Enter **`yes`** when prompted to continue.

```
dmf-controller-1(config)# tunneling
Tunneling is an Arista Licensed feature.
Please ensure that you have purchased the license for tunneling before using this feature.
Enter "yes" (or "y") to continue: y
dmf-controller-1(config)#
```

2. Configure a tunnel by using the ***remote-ip*** as **`0.0.0.0`**.

```
dmf-controller-1(config)# switch main-dc-delivery-sw
dmf-controller-1(config-switch)# gre-tunnel-interface tunnel1
dmf-controller-1(config-switch)# remote-ip 0.0.0.0 === this is to enable wildcard tunnel
dmf-controller-1(config-switch)# gre-key-decap 4097
dmf-controller-1(config-switch)# parent-interface ethernet5
dmf-controller-1(config-switch)# local-ip 192.168.200.50 mask 255.255.255.0 gateway-ip 192.168.200.1
dmf-controller-1(config-switch)# direction receive-only
```

Please refer to the Tunneling Between Data Centers chapter for more information on using L2GRE tunnels in DANZ Monitoring Fabric (DMF).

Show Commands

All CLI show commands for regular interfaces apply to GRE tunnel interfaces. Use the **`show running-config`** command to view the configuration of tunnel interfaces.

Enter the **`show tunnel`** command to view a tunnel interface's configuration parameters and runtime state.

Example

```
dmf-controller-1# show tunnel
# Switch DPID       Tunnel Name Tunnel Status      Direction     Src IP       Dst IP       Parent Name  Loopback
 Name
-|-----------------|-----------|------------------|-------------|------------|------------|------------|--------
-----|
1 DMF-CORE-SWITCH-1 tunnel1     ESTABLISHED_TUNNEL bidirectional 198.82.215.1 216.47.143.1 ethernet5:1
 ethernet6
2 DMF-CORE-SWITCH-2 tunnel1     ESTABLISHED_TUNNEL bidirectional 216.47.143.1 198.82.215.1 ethernet11:3
 ethernet5
3 DMF-CORE-SWITCH-2 tunnel2     ESTABLISHED_TUNNEL bidirectional 192.168.43.1 192.168.42.1 ethernet11:4
 ethernet17
4 DMF-CORE-SWITCH-3 tunnel2     ESTABLISHED_TUNNEL bidirectional 192.168.42.1 192.168.43.1 ethernet6
 ethernet33
```

8.9.4 Encapsulation Types

DANZ Monitoring Fabric (DMF) supports VXLAN tunnel type and Level-2 Generic Routing Encapsulation (L2GRE). The tunnel type is a per-switch configuration, setting the switch pipeline to VXLAN or L2GRE. Once the switch pipeline is set, all tunnels configured on the switch will use the same tunnel type.

The encapsulation type can also be configured or edited from the CLI in configuration mode:

```
Ctrl-1(config)# switch Switch-1
Ctrl-1(config-switch)# tunnel-type
gre Select GRE as the tunnel type of the switch. (default selection)
vxlan Select VxLAN as the tunnel type of the switch.
```

The switch pipeline mode can be viewed from the CLI using the following command:

```
Ctrl-1(config)# show switch
# Switch Name         IP Address                  State     Pipeline Mode
-|------------------|--------------------------|---------|---------------------------------|
1 Switch-1           fe80::d6af:f7ff:fef9:e2b0%9 connected l3-l4-offset-match-push-vlan-vxlan
2 Switch-2           fe80::e6f0:4ff:fe69:6aee%9  connected l3-l4-offset-match-push-vlan
3 Switch-3           fe80::e6f0:4ff:fe78:1ffe%9  connected l3-l4-offset-match-push-vlan-vxlan
Ctrl-1(config)#
```

In the above CLI output, ***Switch-1*** and ***Switch-3*** use the VXLAN tunnel type, as seen in the **`Pipeline Mode`** column. ***Switch-2*** is using the L2GRE tunnel type.

Chapter 9

Integrating vCenter with DMF

This chapter describes integrating VMware vCenter with the DANZ Monitoring Fabric (DMF) and monitoring Virtual Machines (VM) in the vCenter.

Overview

Using SPAN to Monitor VMs

Using ERPSAN to Monitor VMs

Defining a Tunnel Endpoint

Integrate a vCenter Instance

View vCenter Configuration

Integrating vCenter with DMF using Mirror Stack

Create Wildcard Tunnels for VMware vCenter Monitoring

Minimum Permissions for Non-admin Users

Monitor vCenter Traffic by VM Names

Using the Command Line Interface

9.1 Overview

The DANZ Monitoring Fabric (DMF) allows the integration and monitoring of VMs in a VMware vCenter cluster. After integrating a vCenter with the DMF fabric, use DMF policies to select different types of traffic from specific VMs and apply managed services, such as deduplication or header slicing, to the selected traffic.

Currently, DMF supports the following versions of VMware vCenter for monitoring:

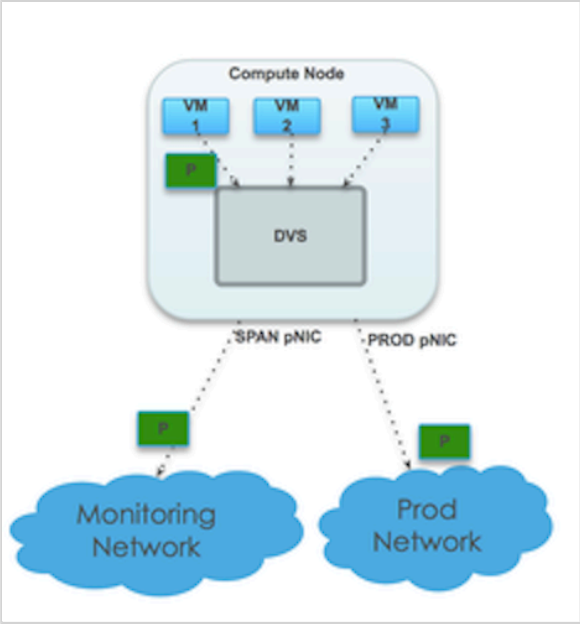
- vCenter Server 7.0.0
- vCenter Server 8.0.0

The DANZ Monitoring Fabric provides two options to monitor a VMware vCenter cluster:

- **Monitoring using span ports**: This method monitors VMware vCenter clustering using a separate monitoring network. The advantage of this configuration is that it has no impact on the production network and has a minimal effect on compute node CPU performance. However, in this configuration, each compute node must have a spare NIC to monitor traffic.

  The following figure illustrates the topology used for local SPAN configuration:

  Figure 9-1: Mirroring on a Separate SPAN Physical NIC (SPAN)

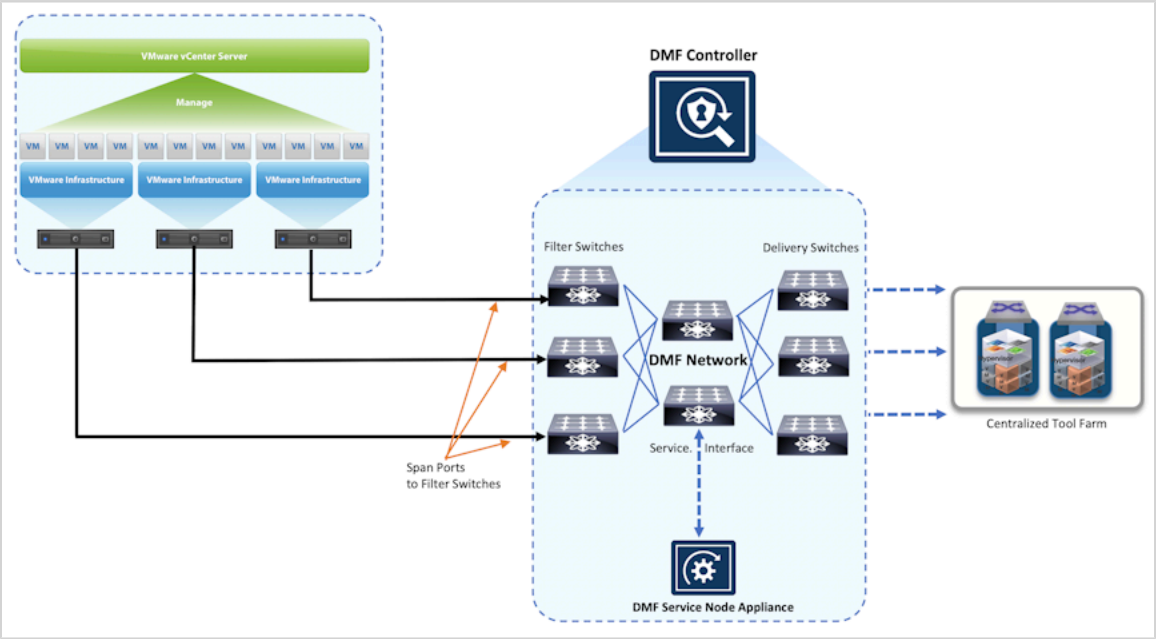
- **Monitoring using ERPAN/L2GRE tunnels**: Use Remote SPAN (ERSPAN) to monitor VMs running on the ESX hosts within a vCenter instance integrated with DMF. ERSPAN monitors traffic to and from VMs anywhere in the network and does not require a dedicated physical interface card on the ESX host. However, ERSPAN can affect network performance, especially when monitoring VMs connected to the DMF Controller over WAN links or production networks with high utilization.

9.2 Using SPAN to Monitor VMs

This section describes the configuration required to integrate the DANZ Monitoring Fabric (DMF) Controller with one or more vCenter instances and to monitor traffic from VMs connected to the VMware vCenter after integration.

The following figure illustrates the topology required to integrate a vCenter instance with the monitoring fabric and deliver the traffic selected by DMF policies to specified delivery ports connected to different monitoring tools.

Figure 9-2: VMware vCenter Integration and VM Monitoring

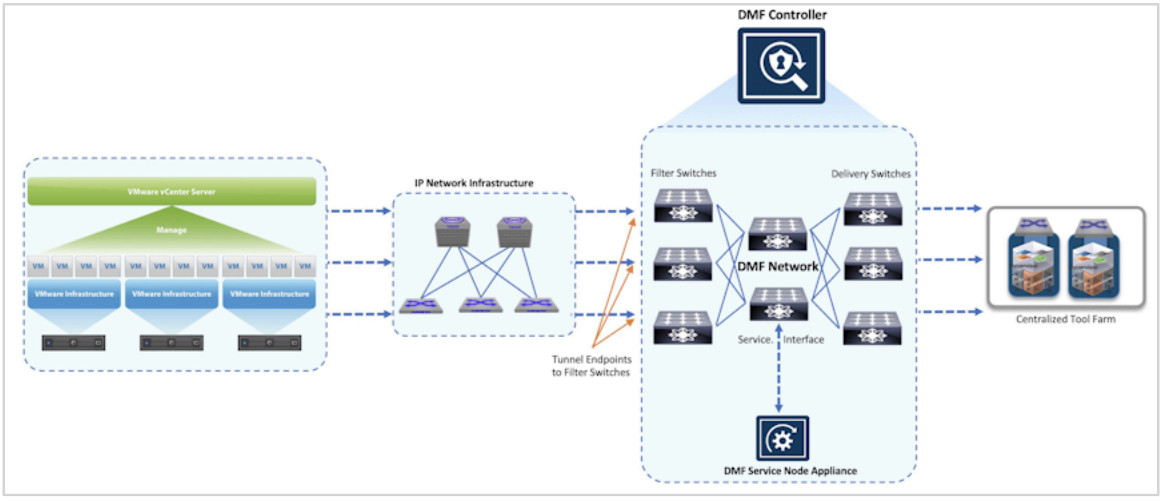


When integrated with vCenter, the DMF Controller uses Link Layer Discovery Protocol (LLDP) to automatically identify the available filter interfaces connected to the vCenter instance.

9.3 Using ERSPAN to Monitor VMs

Use Remote SPAN (ERSPAN) to monitor VMs running on the ESX hosts within a VMware vCenter instance integrated with the DANZ Monitoring Fabric (DMF). ERSPAN monitors traffic to and from VMs anywhere in the network and does not require a dedicated physical interface card on the ESX host. However, ERSPAN can affect network performance, especially when monitoring VMs connected to the DMF Controller over WAN links or production networks with high utilization.

Figure 9-3: Using ERSPAN to Monitor VMs



The procedure for deploying ERSPAN is similar to SPAN but requires an additional step to define the tunnel endpoints used on the DMF network to terminate the ERSPAN session.

9.3.1 Configuration Summary for vCenter Integration

The following procedure summarizes the high-level steps required to integrate the vCenter and monitor traffic to or from selected VMs:

1. (For ERSPAN only) Define the tunnel endpoint.

   Identify a fabric interface connected to the vCenter instance for the tunnel endpoint by entering the **`tunnel-endpoint`** command in config mode. To define the tunnel endpoint, refer to the ***Defining a Tunnel Endpoint*** section.
2. Provide the vCenter address and credentials.

   The vSphere extension on the DANZ Monitoring Fabric (DMF) Controller discovers an inventory of VMs and the associated details for each VM.
3. Select the VMs to monitor on the DMF Controller.

   The DMF Controller uses APIs to invoke the vSphere vCenter instance.

   vSphere calls the DVS to create a SPAN session. The preferred option is to SPAN on a separate physical NIC. However, the option exists to also use ERSPAN by tunneling to the remote interface.
4. Create policies in DMF to filter, replicate, process, and redirect traffic to tools.

   When using tunnels with ERSPAN, DMF terminates the tunnels using the specified tunnel endpoint. A DMF policy for monitoring VM traffic using a SPAN session must include the required information regarding the vCenter configuration. All match conditions, including User-Defined ofFsets (UDFs), are supported.

   The policy for selecting VM traffic to monitor is similar to other DMF policies, except that the filtering interfaces are orchestrated automatically (filter interfaces are auto-discovered and cannot be specified manually). All managed-service actions are supported.

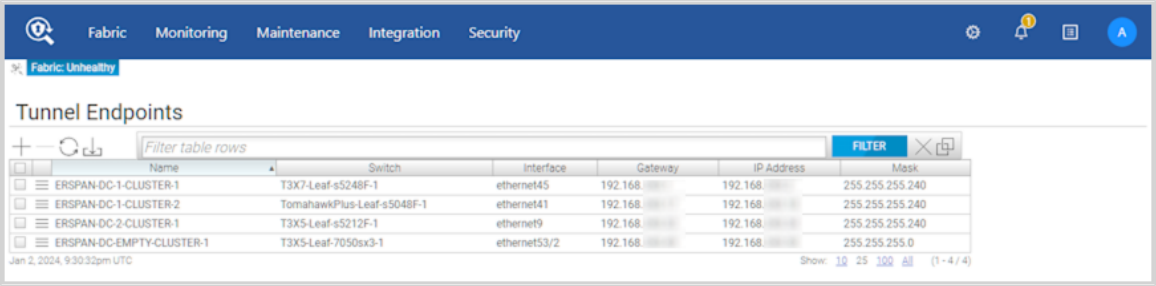
9.4 Defining a Tunnel Endpoint

Predefine the tunnel endpoints for creating tunnels when monitoring VMware vCenter traffic using either the GUI or the CLI.

GUI Procedure

To manage tunnel endpoints in the GUI, select **Monitoring > Tunnel Endpoints**.

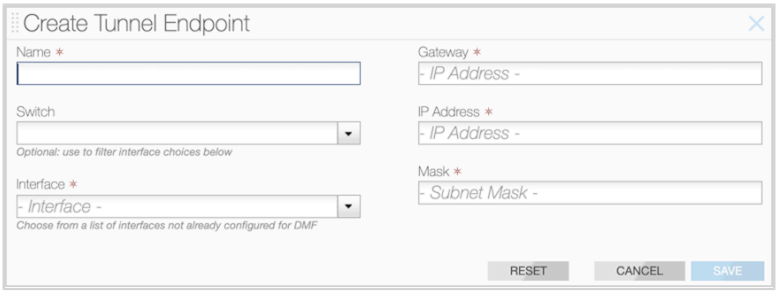
Figure 9-4: Monitoring > Tunnel Endpoints



This page lists the tunnel endpoints that are already configured and provides information about each endpoint.

To create a new tunnel endpoint, select the provision (**+**) control in the **Tunnel Endpoints** table.

Figure 9-5: Create Tunnel Endpoint



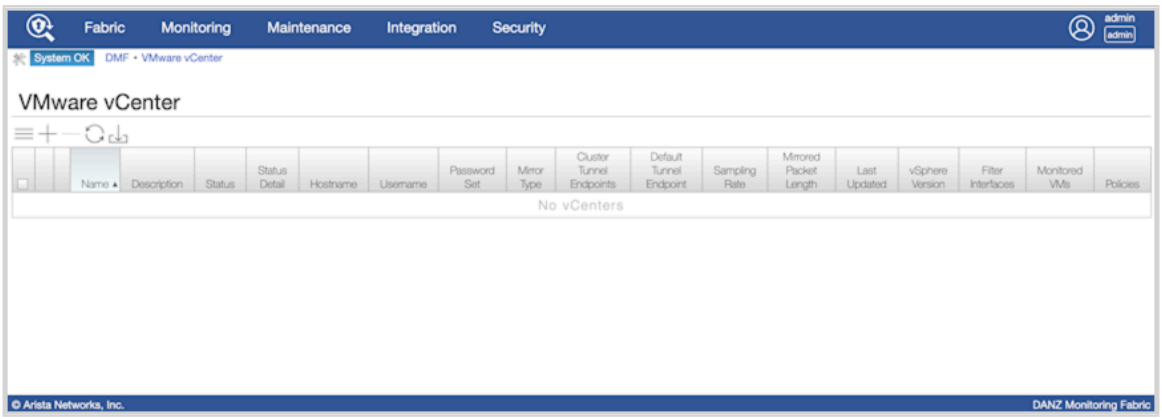
To create the tunnel endpoint, enter the following information and select **Save**:

- **Name**: Type a descriptive name for the endpoint.
- **Switch**: Select the DMF switch from the selection list for the configured endpoint interface.
- **Interface**: Select the interface from the selection list for the endpoint.
- **Gateway**: Type the address of the default gateway.
- **IP Address**: Type the endpoint IP address.
- **Mask**: Type the subnet mask for the endpoint.

9.5 Integrate a vCenter Instance

To integrate a vCenter instance with DANZ Monitoring Fabric (DMF) to begin monitoring VMs, select **Integration > vCenter** from the **DMF** menu bar.

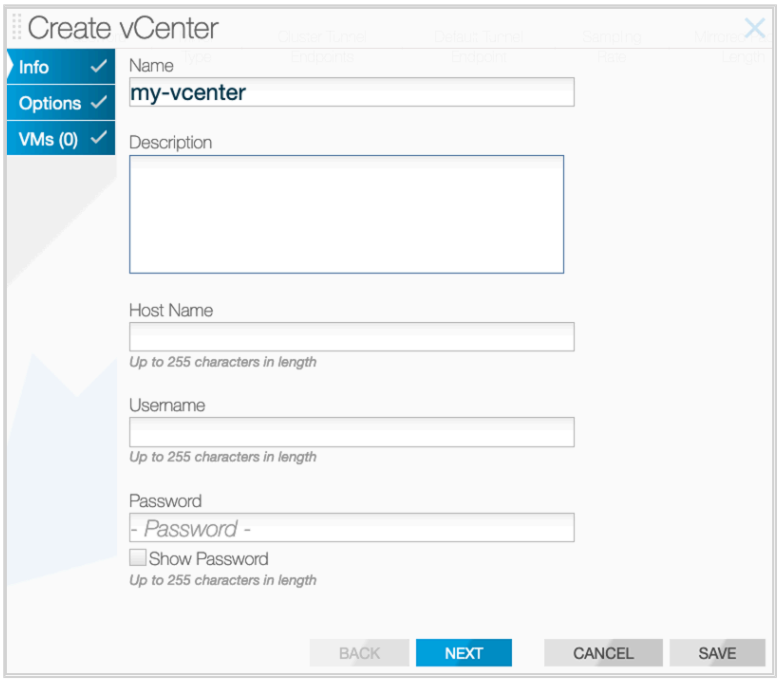
Figure 9-6: Integration > vCenter



This page displays information about the vCenter instances integrated with DMF. To add a vCenter instance for integration with DMF, perform the following steps:

1. Select the provision control (**+**) in the table.

   Figure 9-7: Create vCenter: Info



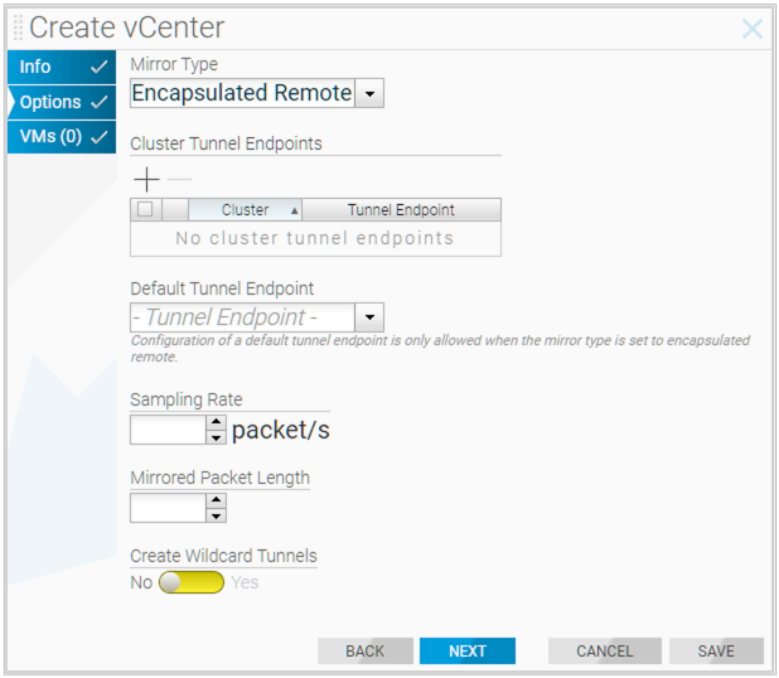
2. Type an alphanumeric identifier for the vCenter instance, and (optionally) add a description in the fields provided.
3. Identify the vCenter hostname to be integrated.

4. Enter the vCenter username and password for authenticating to the vCenter instance.

   These credentials are used by the DMF Controller when communicating with the vCenter host.

5. Select **Next**.

Figure 9-8: Create vCenter: Options (page 2)



This page defines the mirror type as SPAN or ERSPAN. When selecting ERSPAN, the following additional fields complete the ERSPAN configuration:

- **Cluster Tunnel Endpoints** (optional)
- **Default Tunnel Endpoint** (required)
- **Sampling Rate** (optional)
- **Mirrored Packet Length** (optional)
- **Create Wildcard Tunnels** (optional)

Use **Cluster Tunnel Endpoints** to specify a common tunnel endpoint for all the ESXi hosts in the cluster. Use **Default Tunnel Endpoint** to specify a common tunnel endpoint for all the ESXi hosts regardless of the cluster. When configuring both cluster and default tunnel endpoints, all hosts in clusters form tunnels using the cluster-specific configuration, and all the other hosts that are not a part of any cluster use the default configuration to form tunnels.

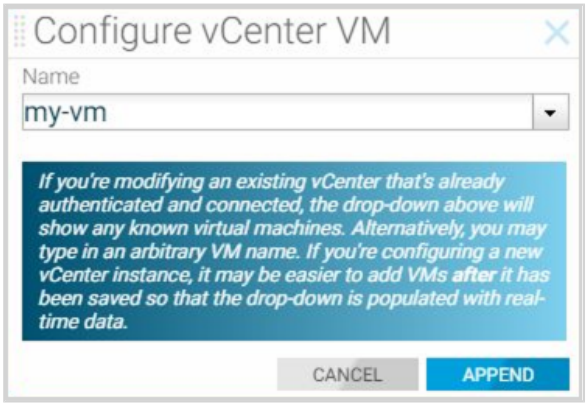
6. Select **Next**.

Figure 9-9: Create vCenter/VMs



7. To add a VM for monitoring, select the provision control (**+**).

Figure 9-10: Configure vCenter VM



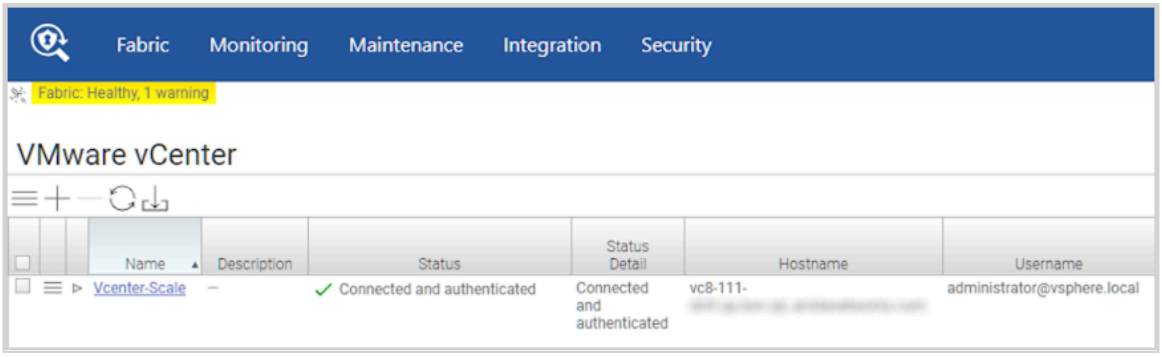
Select VMs from the selection list after integrating vCenter and discovering the VMs, or manually add the VM hostname.

8. After identifying the VM to monitor, select **Append**.
9. On the **VMs** of the **Create vCenter** dialog, select **Save**.

9.5.1 Using a vCenter Instance as the Traffic Source in a DMF Policy

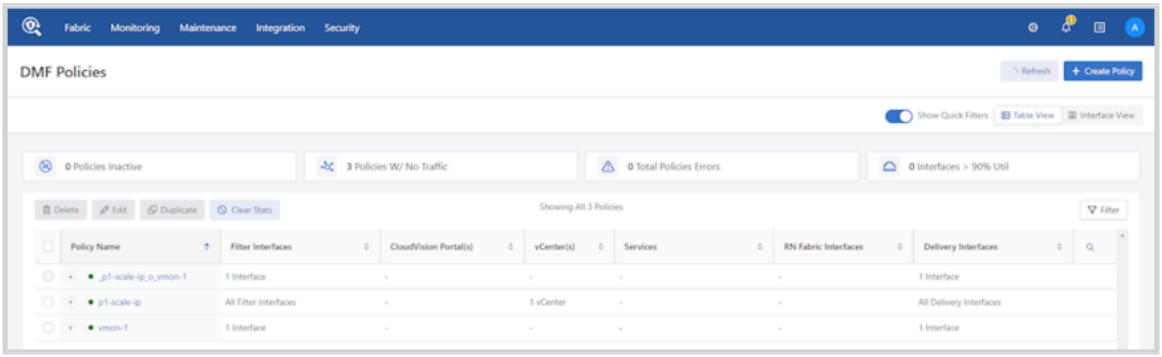
To identify a vCenter instance integrated with the DANZ Monitoring Fabric (DMF) Controller as the traffic source for a DMF policy, select the **VMware vCenter** tab on the **Integration** page. Locate the vCenter instance name.

Figure 9-11: VMware vCenter Name



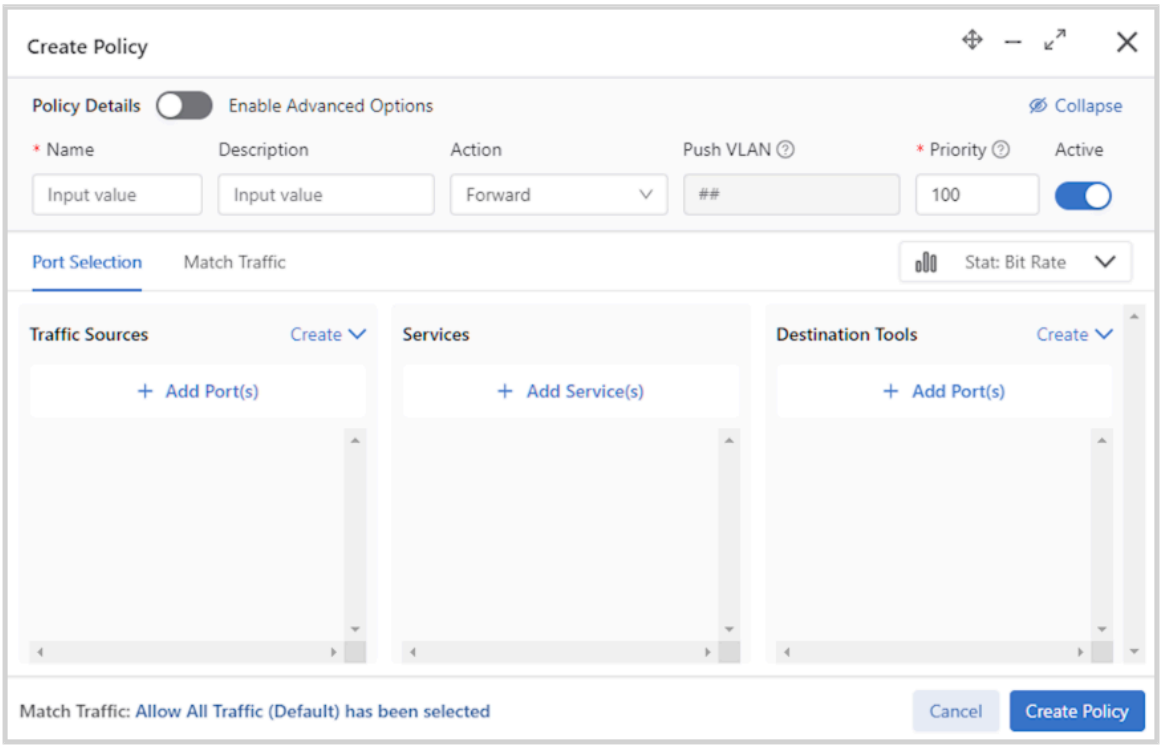
Proceed to the **Monitoring > Policies** page.

Figure 9-12: DMF Policies



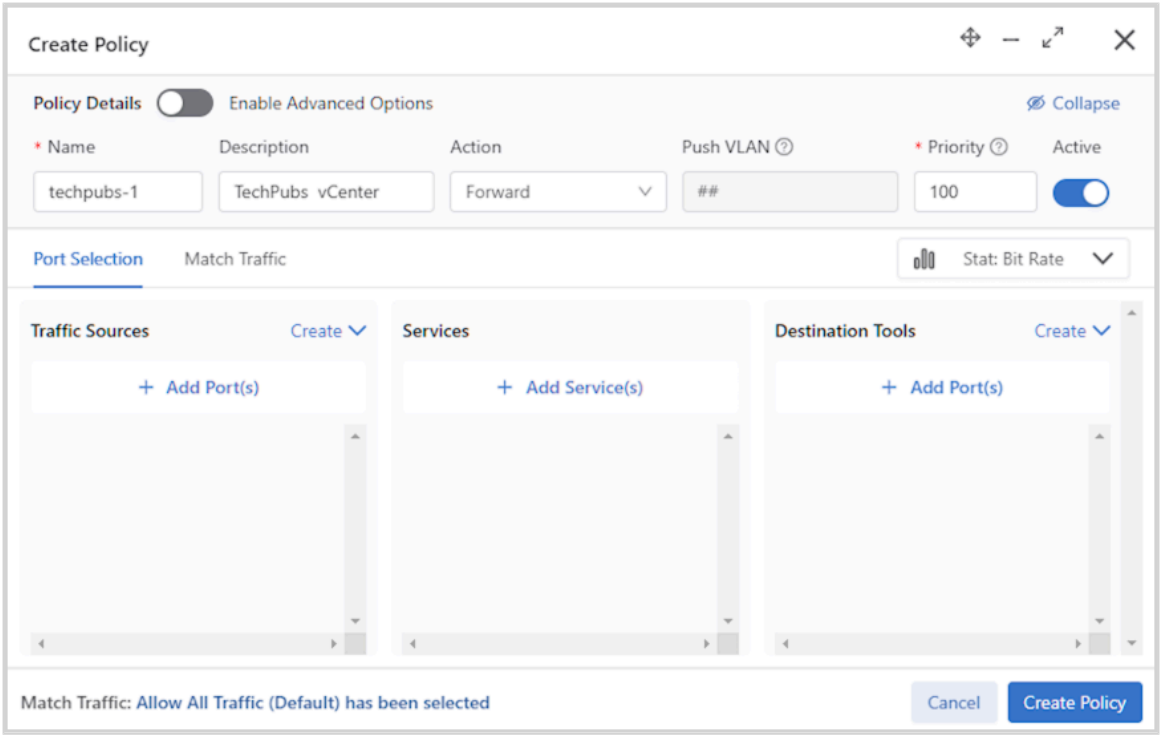
Select **+ Create Policy** to add a policy.

Figure 9-13: Create Policy



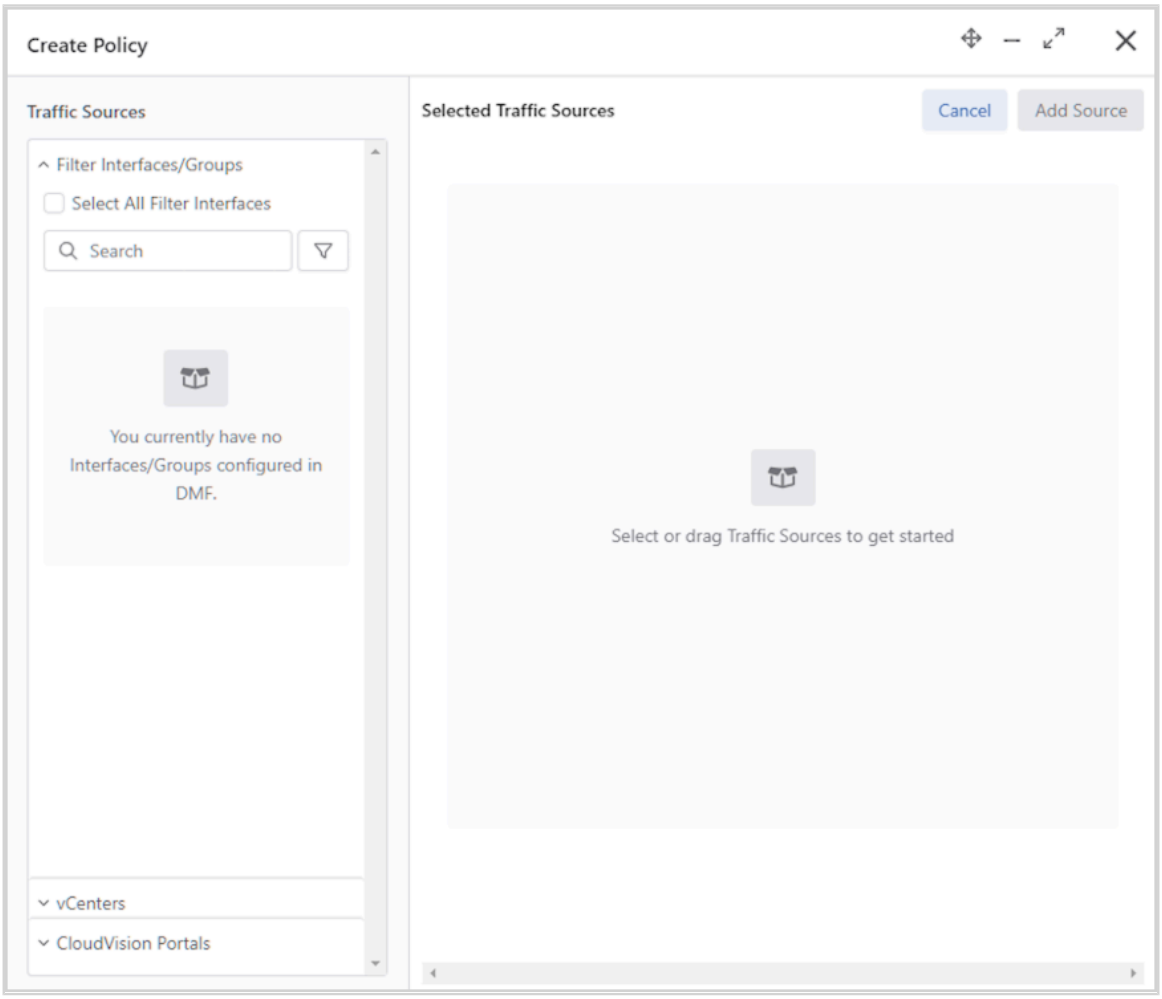
Enter a **Name** and **Description** for the vCenter policy. From the **Traffic Sources** column, select **+ Add Ports(s)**.

Figure 9-14: Traffic Sources - Add Ports



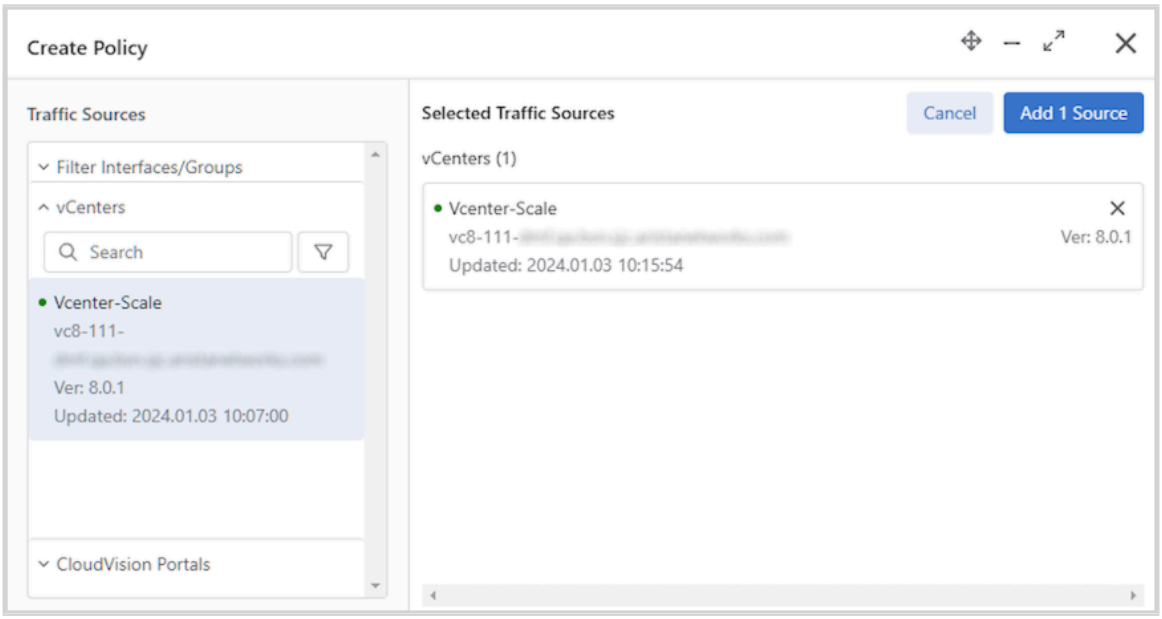
Select **vCenters**.

Figure 9-15: vCenters



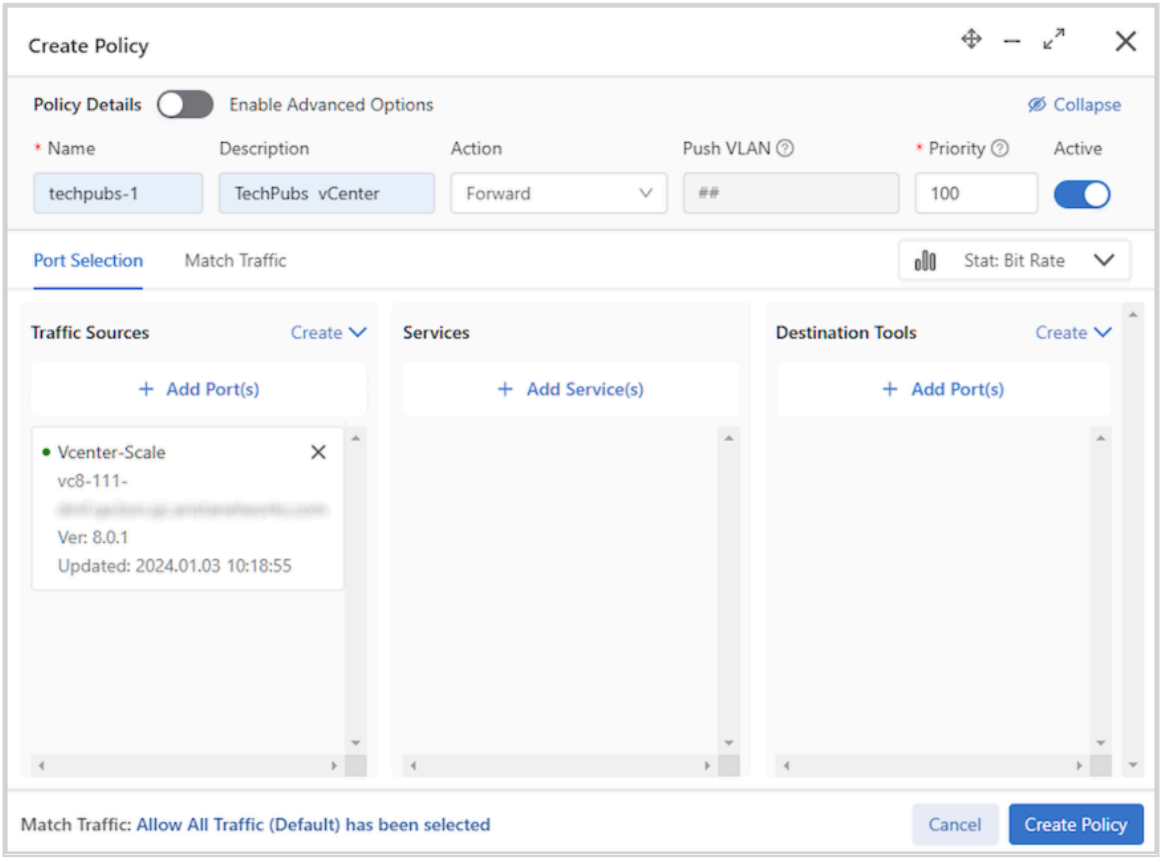
Available vCenter instances display. Select the required vCenter instance which then appears in the Selected traffic Sources panel.

Figure 9-16: vCenter Instance



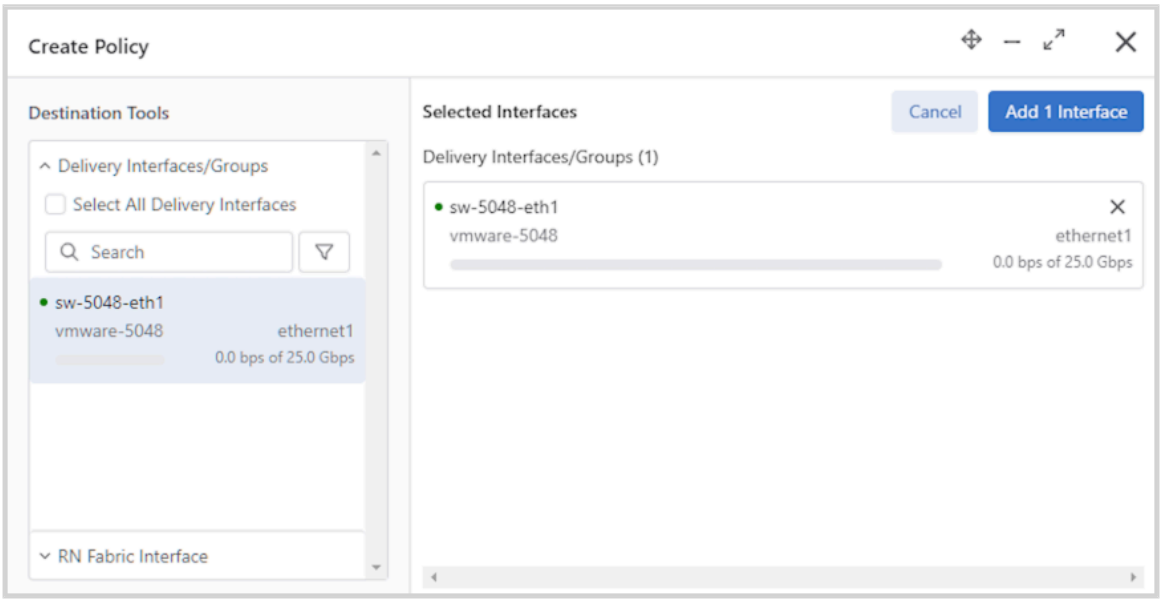
Select **Add 1 Source**. The vCenter instance appears in the **Traffic Sources** column.

Figure 9-17: vCenter Traffic Sources



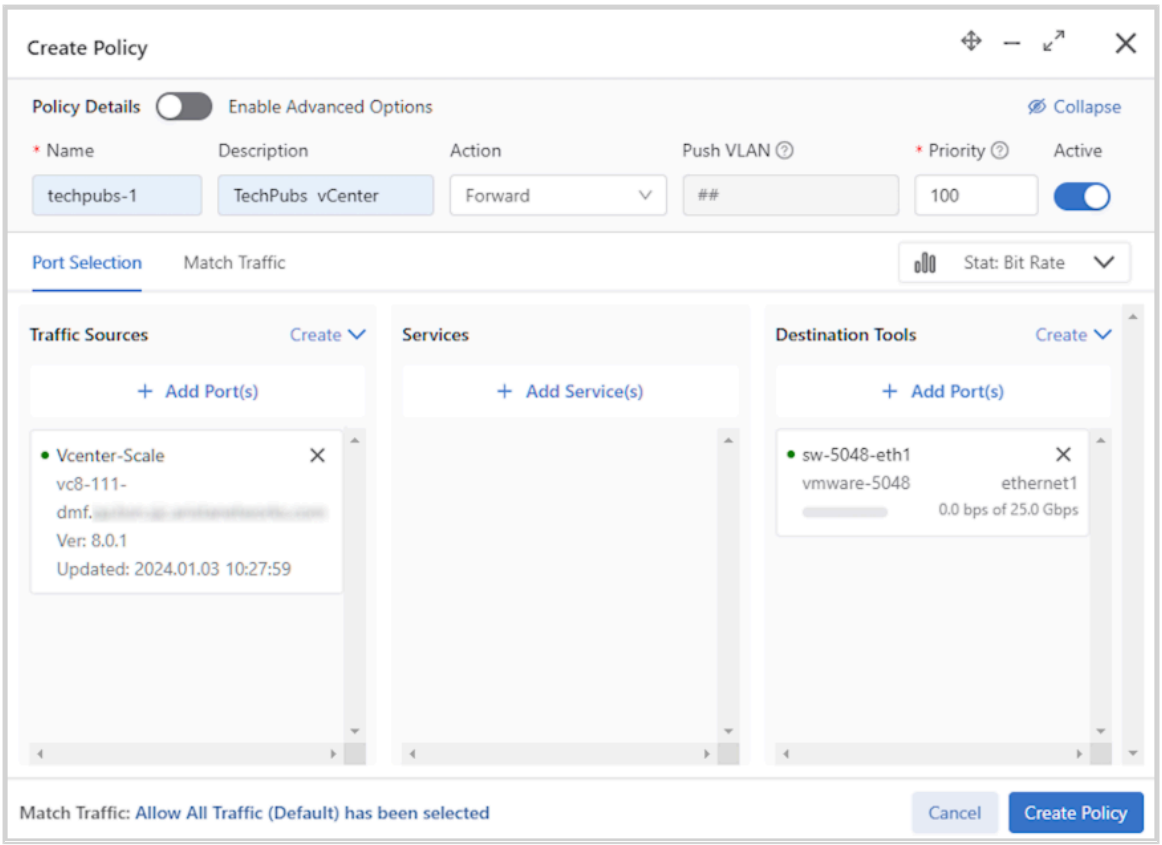
From the **Destination Tools** column, select **+ Add Ports(s)**. Select the interface under **Destination Tools**.

Figure 9-18: Destination Tools - Add Ports



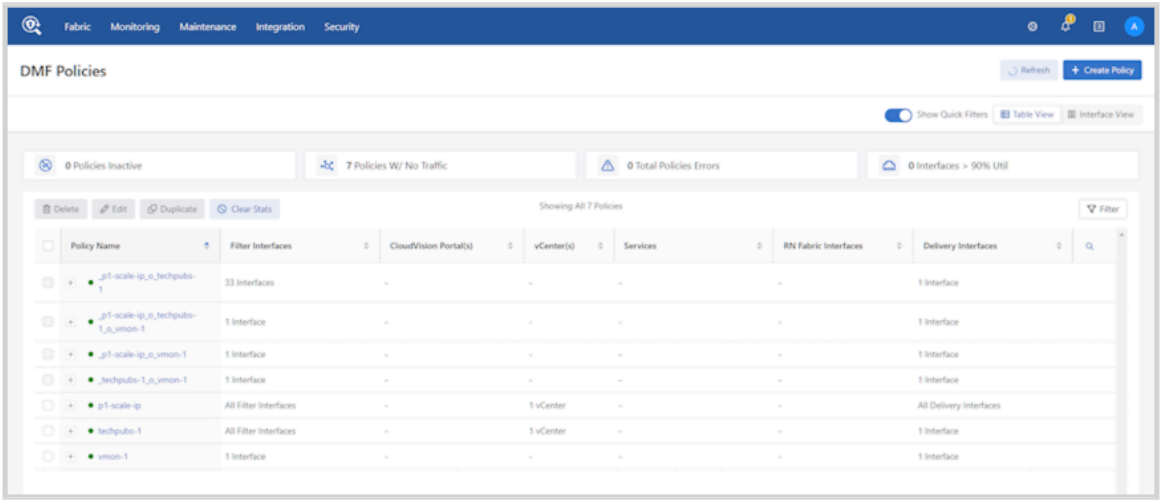
Select **Add 1 Interface**. The interface appears under the **Destination Tools** column.

Figure 9-19: Add Interface



Select **Create Policy**. The new vCenter policy appears in the DMF Policies dashboard.

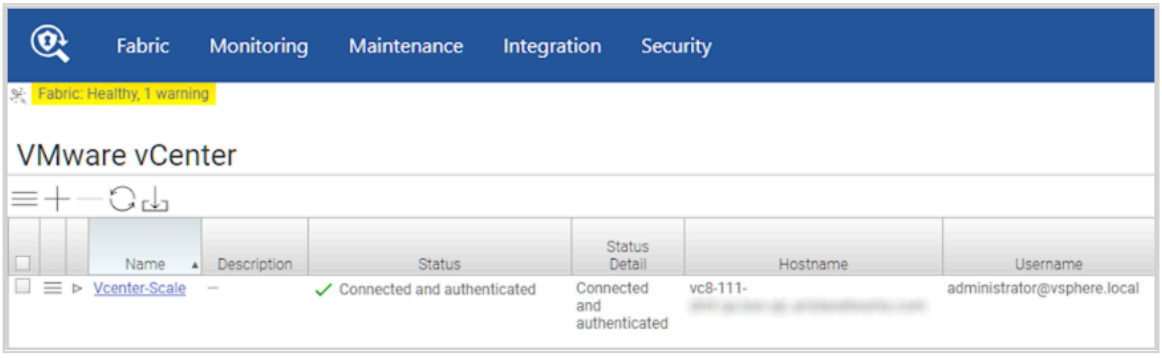
Figure 9-20: Create vCenter Policy



9.6 View vCenter Configuration

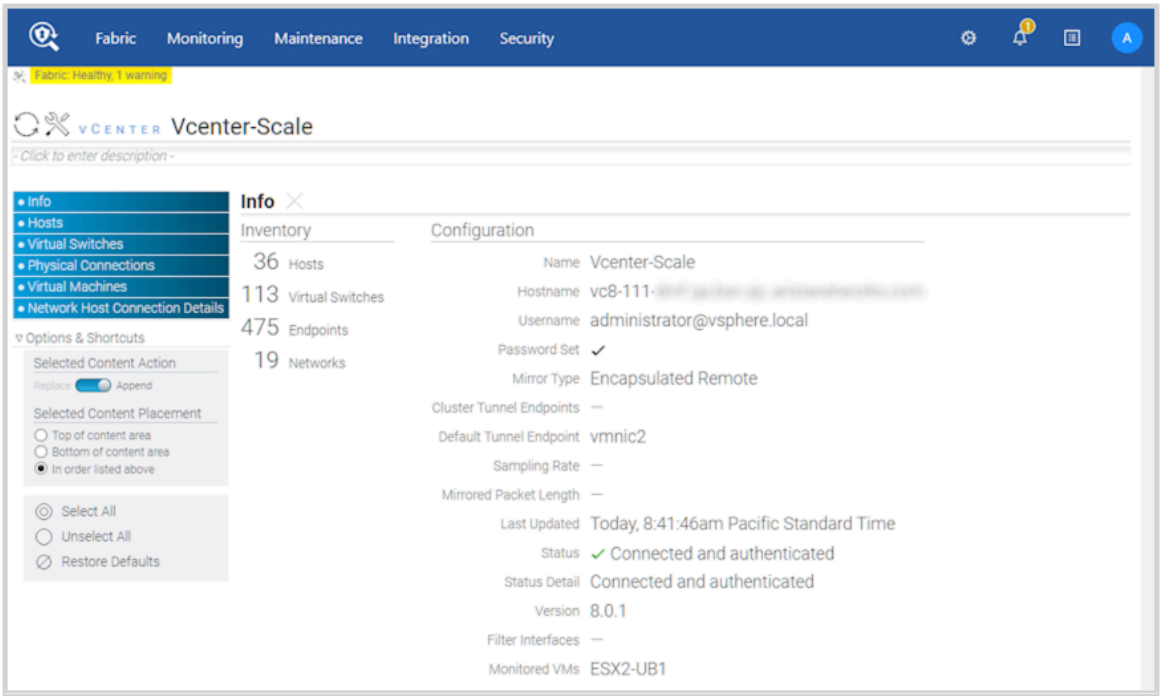
After integrating a vCenter instance, select the link in the **Name** column in the **vCenter** table to view vCenter activity.

Figure 9-21: VMware vCenter Instance Name



DANZ Monitoring Fabric (DMF) displays the **vCenter Info** page.

Figure 9-22: VMware vCenter Configuration



The **Info** page displays information about the configuration of the vCenter instance. To view information about vCenter resources, scroll down to the following sections:

- **Hosts**
- **Virtual Switches**
- **Physical Connections**
- **Virtual Machines**
- **Network Host Connection Details**

Figure 9-23: Hosts, Virtual Switches, and Physical Connections

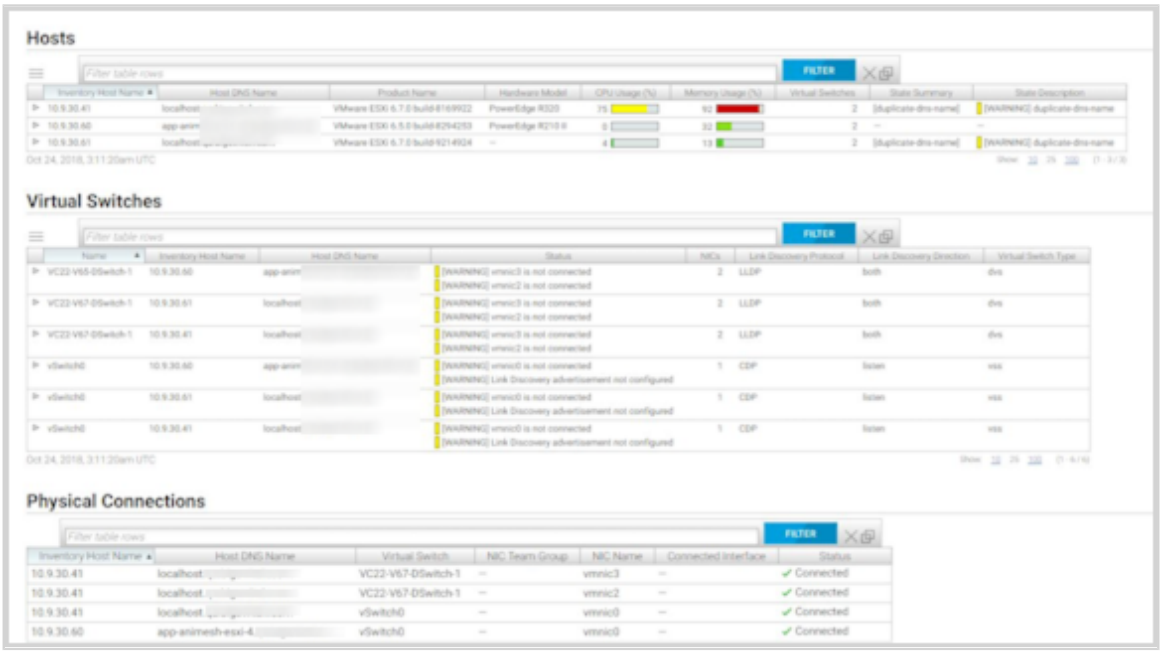
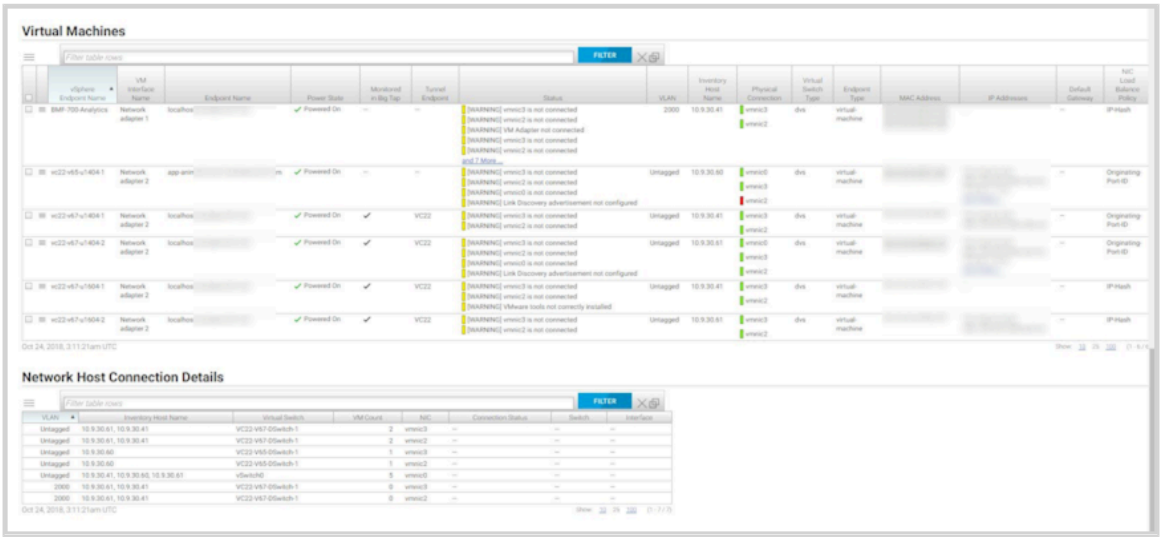


Figure 9-24: Virtual Machines and Network Host Connection Details



9.7 Integrating vCenter with DMF using Mirror Stack

DANZ Monitoring Fabric (DMF) vCenter integration supports mirroring from vCenter hosts using the default TCP/IP stack. However, this can result in traffic drops and affect production traffic since mirror traffic can conflict with production traffic. DMF vCenter integration with Mirror Stack provides the functionality to use the mirror TCP/IP stack for mirror sessions. Mirror stack in the ESXi host allows decoupling the traffic and keeps the production traffic unaffected.

vCenter configurations in DMF will use a mirror stack by default; however, if upgrading from previous DMF versions, the already configured vCenter will be set to use the default TCP/IP stack.

Platform Compatibility

vCenter integration with Mirror Stack requires an **extra** NIC on the ESXi host with following versions:

- vCenter Server 7.0.x
- vCenter Server 8.0.x

9.7.1 vCenter Configuration

DMF vCenter integration with Mirror Stack requires a mirror stack configuration on the ESXi host and vCenter. Perform the following steps to configure the mirror stack on vCenter.

Repeat the steps for each ESXi host containing VMs to be monitored.

1. Enable the mirror stack in the ESXi host if not already enabled.
   a. Use the **`esxcli network ip netstack list`** command to review the current network stacks.

```
[root@ESX33:~] esxcli network ip netstack list
defaultTcpipStack
   Key: defaultTcpipStack
   Name: defaultTcpipStack
   State: 4660

mirror
   Key: mirror
   Name: mirror
   State: 4660
```

   To view the TCP/IP configuration from vCenter UI, navigate to **Host** > **Configure** > **TCP/IP**.

   Figure 9-25: TCP/IP Configuration

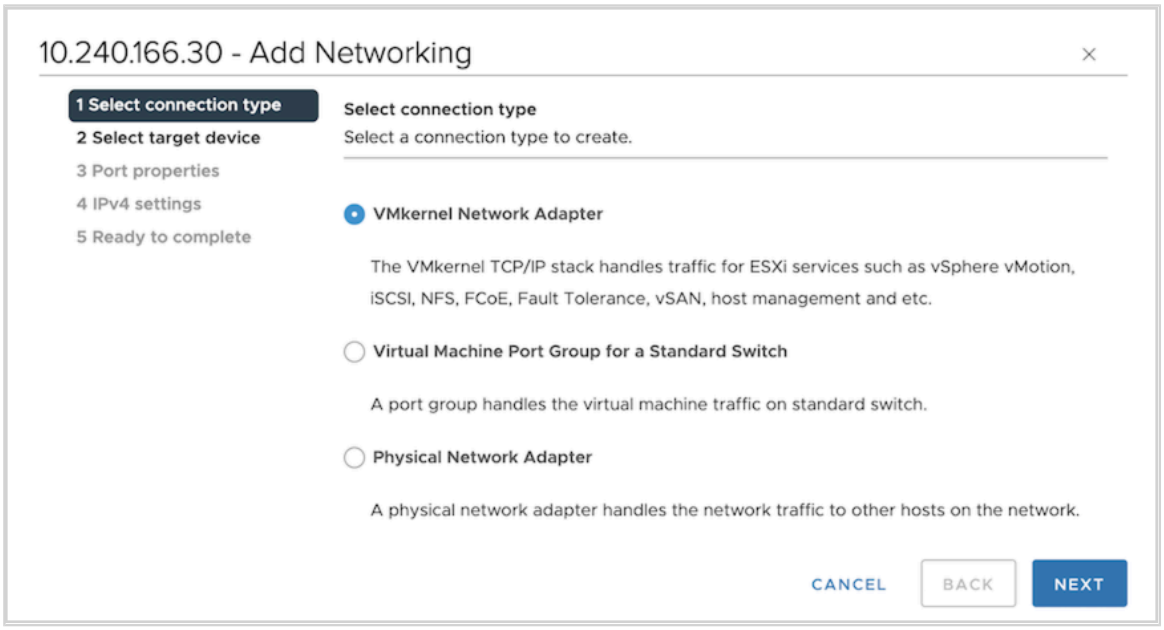


   b. If the mirror stack is not configured, use the **`esxcli network ip netstack set -N mirror`** command to enable it.

> **Note:** The **`mirror`** setting is required to enable the Mirror TCP/IP stack and DMF integration.

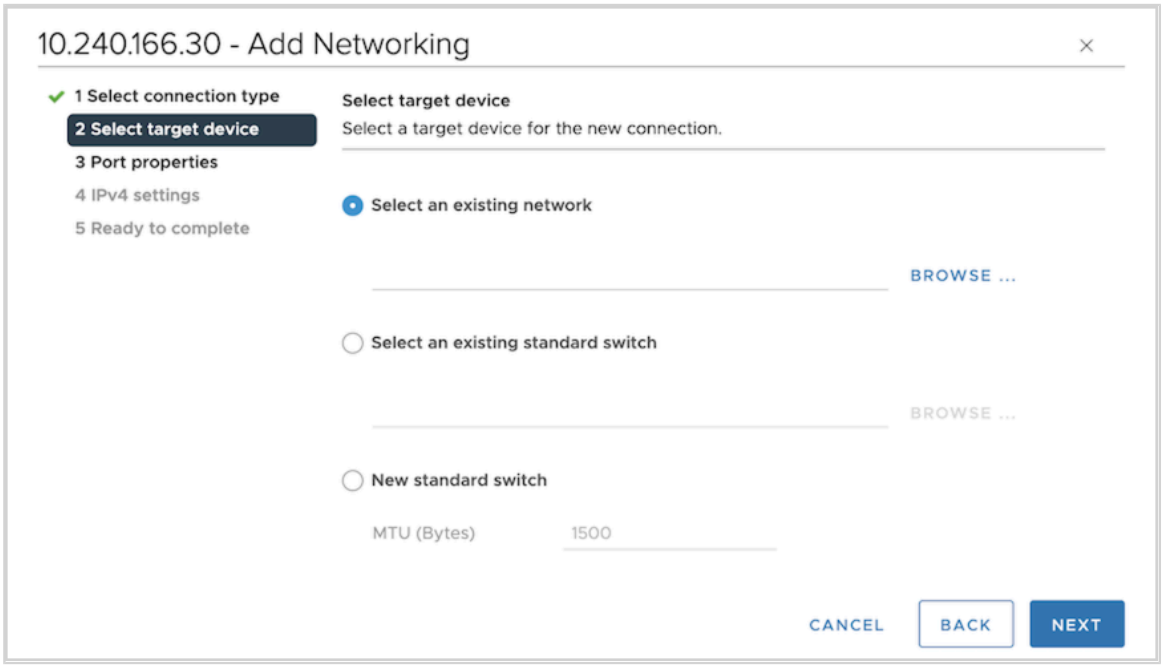
2. From vCenter create a VMkernel adapter with the mirror stack.

Figure 9-26: VMkernel Network Adapter



Select the appropriate network using the **Browse** option.

Figure 9-27: Browse



Select **Next** and select **Port properties** and choose **mirror**.

Figure 9-28: Port Properties - Mirror

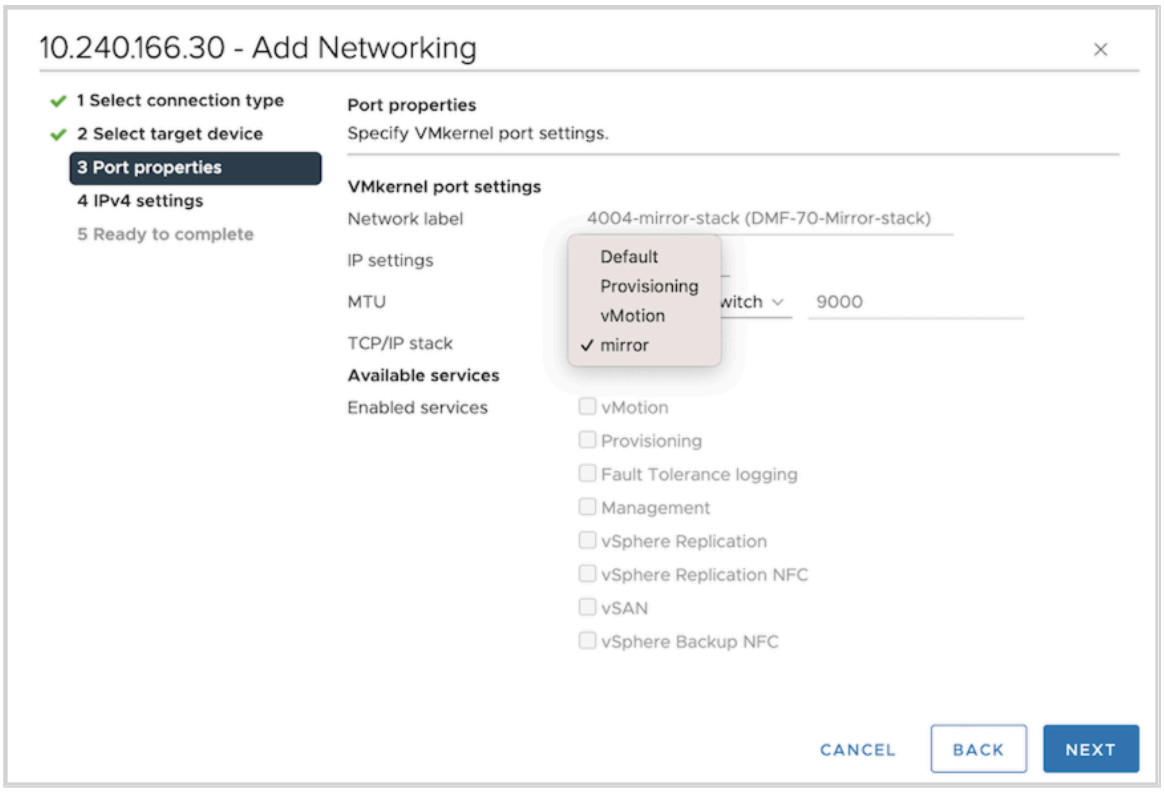
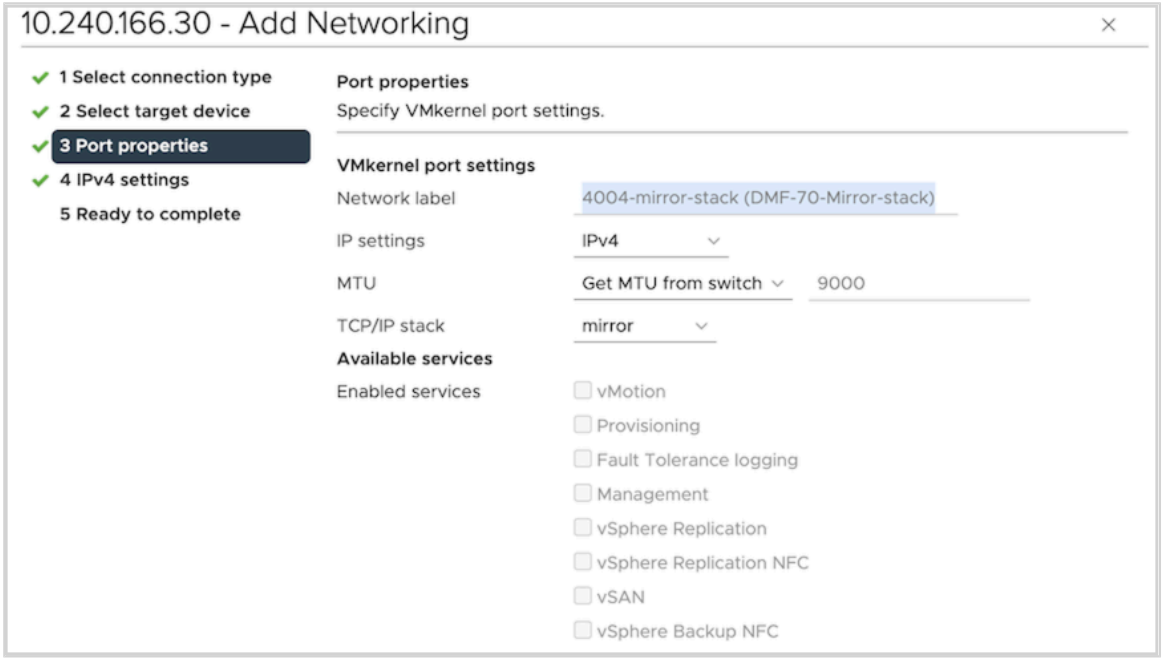
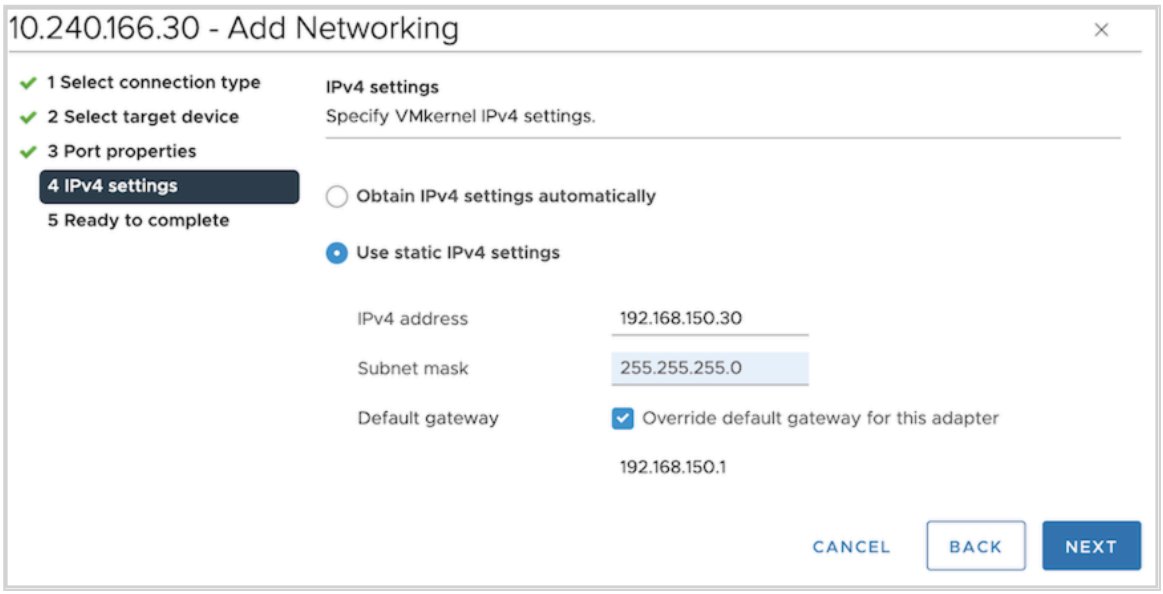


Figure 9-29: Mirror Stack Added



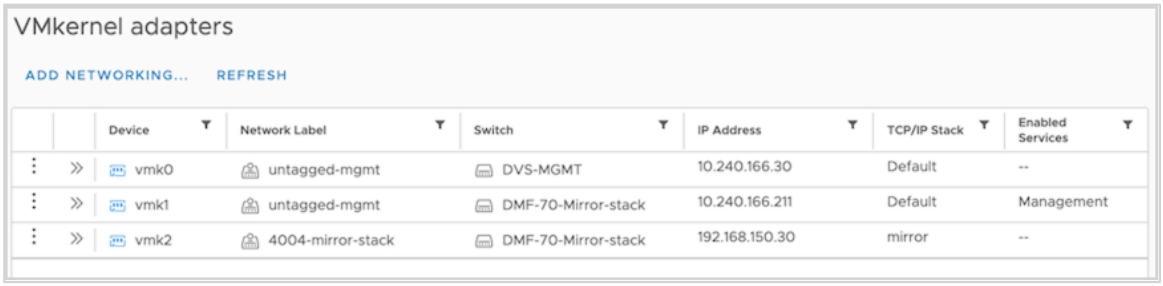
Add the **IPv4 address** and the **Default gateway** address according to your local network requirements.

Figure 9-30: IP Address and Gateway Address



Select **Next**.

Figure 9-31: VMkernel Adapters



3. Based on the networking requirements, configure the default gateway of the mirror stack in the host's TCP/IP configuration or a static route entry in the ESXi host to the DMF tunnel endpoint. The following example illustrates adding a static route entry to the DMF tunnel endpoint.

```
[root@ESX33:~] esxcli network ip route ipv4 add -n 192.168.200.0/24  -g 192.168.150.1 -N mirror

[root@ESX31:~] esxcli network ip route ipv4 list -N mirror
Network        Netmask        Gateway        Interface  Source
-------------  -------------  -------------  ---------  ------
192.168.150.0  255.255.255.0  0.0.0.0        vmk2       MANUAL
192.168.200.0  255.255.255.0  192.168.150.1  vmk2       MANUAL
```

4. Navigate to **Configure** > **TCP/IP Configuration** > **Select mirror stack** > **IPv4 Routing Table** to view the routes.

Figure 9-32: TCP/IP Configuration & IPv4 Routing Table

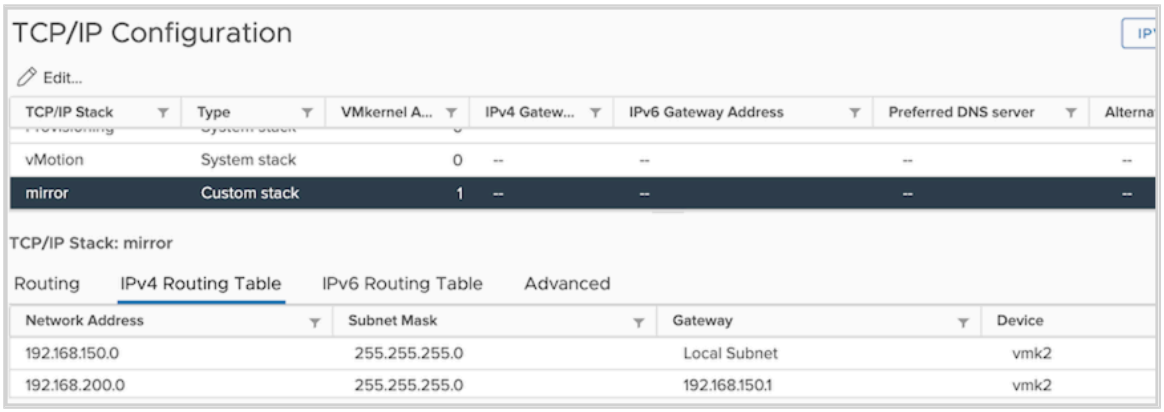
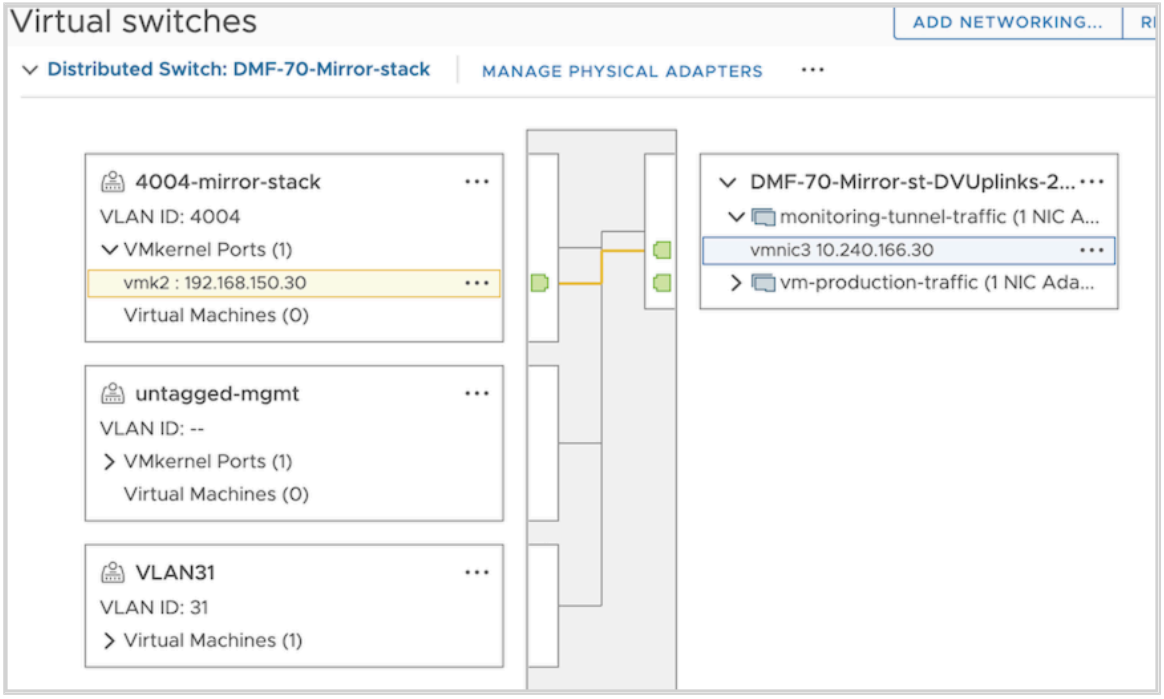


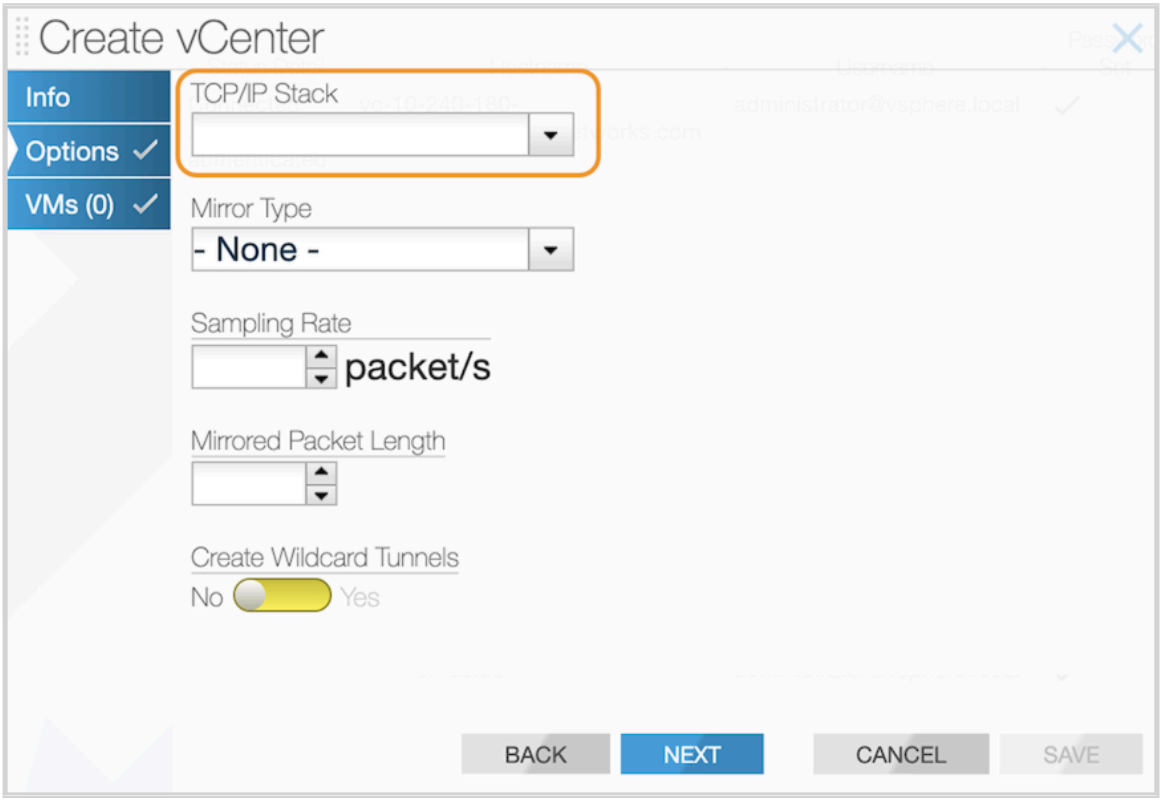
Figure 9-33: Virtual Switch



9.7.2 Configuring DMF

To configure TCP/IP Stack, navigate to **Integration** > **VMware vCenter**. While adding or editing a vCenter configuration, select the appropriate choice using **TCP/IP Stack**. **Default Stack** and **Mirror Stack** are the options.

Figure 9-34: Create vCenter TCP/IP Stack



Attention: Encapsulated Remote mirroring with Default Stack is **not** recommended. Use **Mirror Stack** for optimal performance.

Refer to the CLI **`show`** commands to view the `tcp-ip-stack` configuration. In addition, use the **`show fabric errors`** and **`show fabric warnings`** commands to troubleshoot and verify that everything is functioning as expected.

9.7.3 Limitations

- A port mirroring session remains on the original distributed virtual switch (DVS) when a VM migrates between DVSs.
- Port mirroring sessions will persist on the DVS if a VM is renamed in vCenter while being monitored by DMF.
- DMF cannot create a port mirroring session in the DVS if a conflicting session with the same VM exists in the DVS. This is not a limitation in vCenter 7.
- When using mirror stack configuration in DMF, mirror sessions may still be created on the DVS for the ESXi host that doesn't have a mirror stack configuration. This will result in no traffic being mirrored from the VM.
- Auto-generated filter interfaces by vCenter integration should not be deleted from the policy. If they are deleted manually from the policy, they will not be automatically re-added.
- DMF cannot monitor VMkernel adapters.

9.7.4 Resources

- Create a Port Mirroring Session - https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-networking/GUID-68B5DD45-DD3F-4E9B-A6CD-BE97026A846A.html
- Encapsulated Remote Mirroring (L3) Source - https://kb.vmware.com/s/article/67973

9.8 Create Wildcard Tunnels for VMware vCenter Monitoring

The current implementation of VMware vCenter creates one tunnel interface from every ESXi host to DMF.

Using a wildcard tunnel on DMF for VMware vCenter reduces the number of tunnels created.

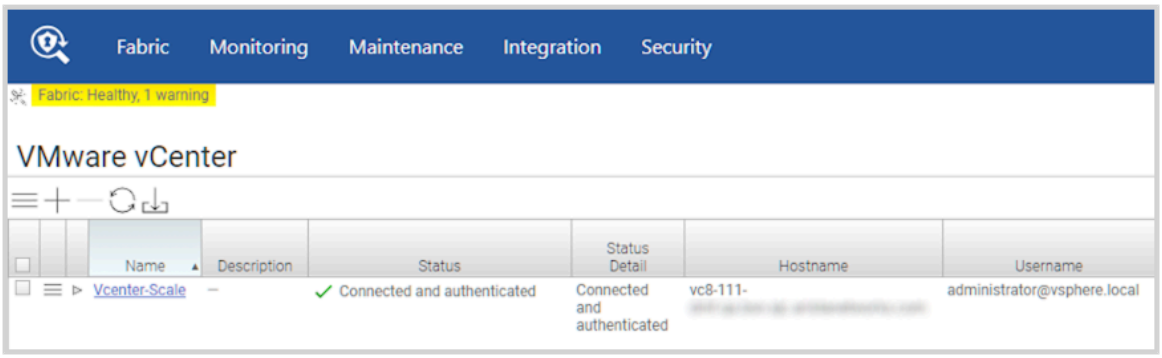
Platform Compatibility

This feature is only compatible with switches that support wildcard tunneling.

Use the DANZ Monitoring Fabric (DMF) GUI to create wildcard tunnels as outlined below.

Navigate to the **Integration > VMware vCenter** page.

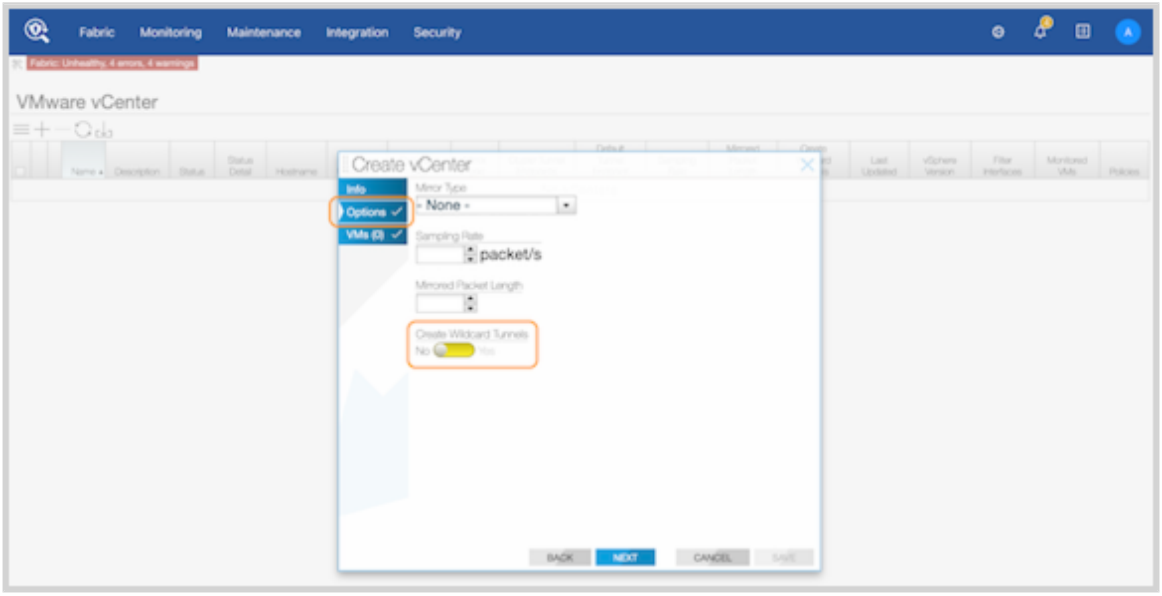
Figure 9-35: VMware vCenter Add/Edit



Select the **Menu** icon.

As part of the **Options** step of the **Add/Edit vCenter** workflow, enable wildcard tunnels using the **Create Wildcard Tunnels** toggle input. By default, the feature is disabled.

Figure 9-36: VMware vCenter Create vCenter Options



9.8.1 Limitations

Select Broadcom® switch ASICs support wildcard tunnels; ensure your switch model supports this feature before configuring it for vCenter.

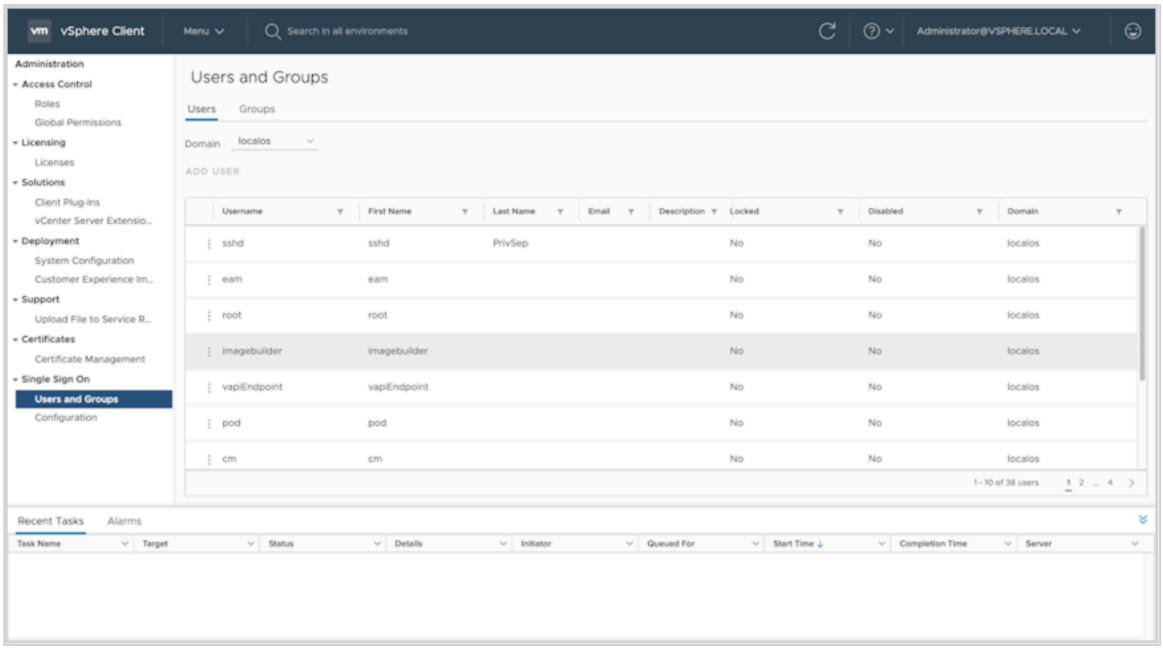
Please refer to the Platform Compatibility section for more information.

9.9 Minimum Permissions for Non-admin Users

For a non-admin user to add, remove, edit, or monitor a vCenter via the DANZ Monitoring Fabric (DMF), the privilege level assigned to the non-admin user is VSPAN operation. To assign VSPAN operation privileges to a user, perform the following steps:

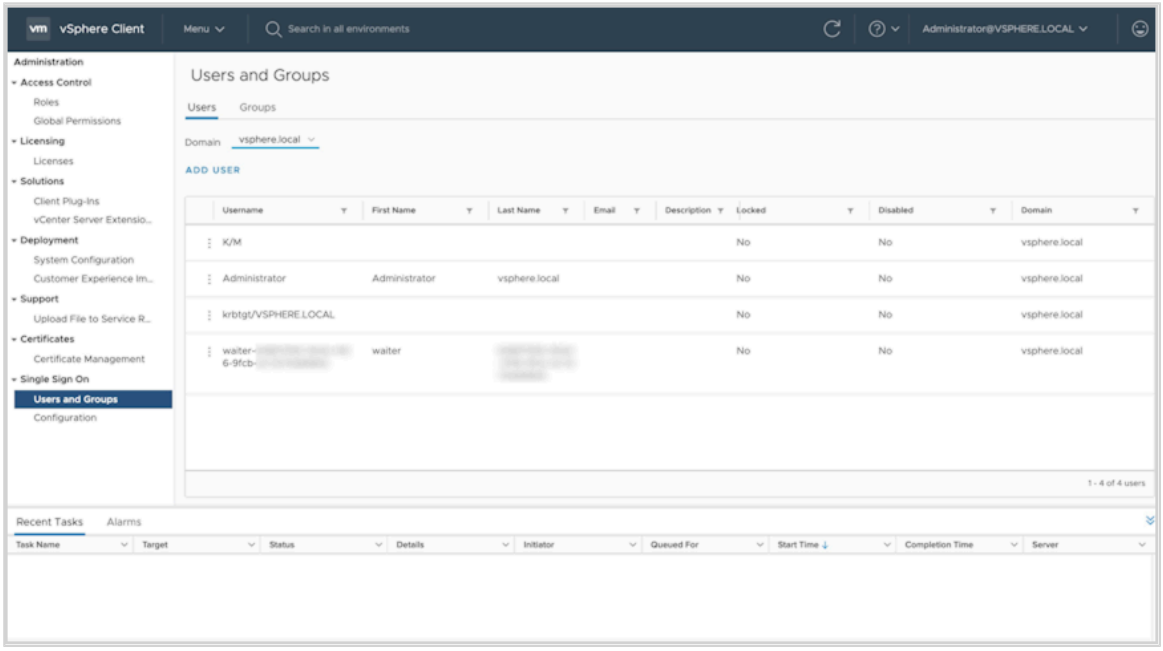
1. From the vCenter GUI, navigate to **Menu > Administration**.
2. Once on the page, select the **Users and Groups** link in the navigation bar on the left.

Figure 9-37: Users and Groups



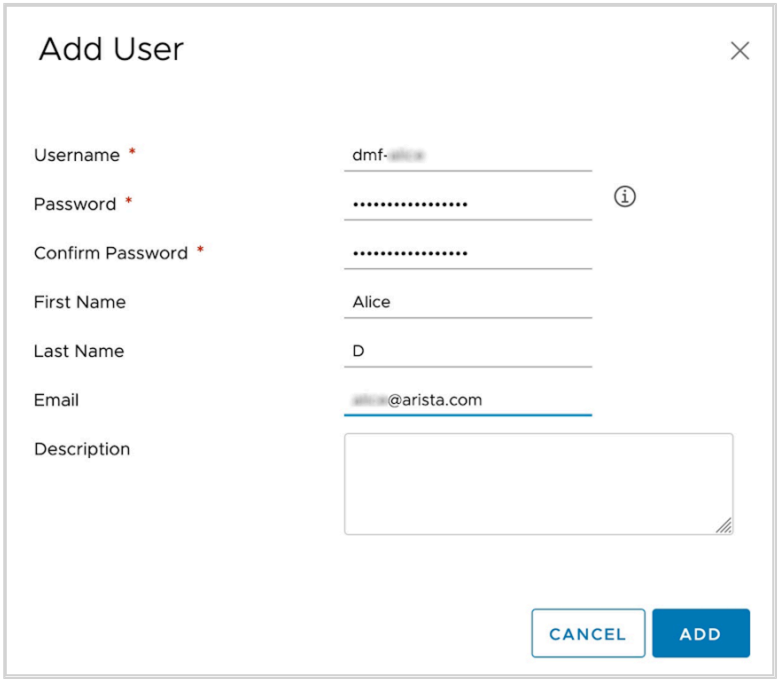
3. Select the **Users** tab and ensure the appropriate domain is selected (in this case, the domain is `vsphere.local`).

Figure 9-38: Domain Selection



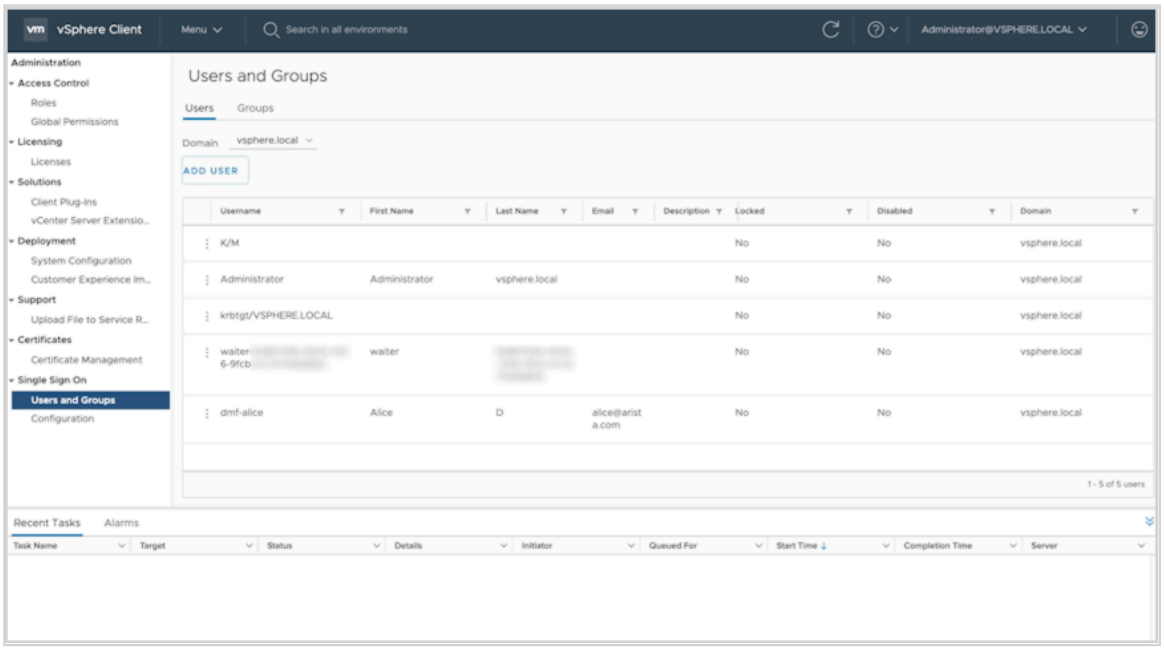
4. Next, select the **ADD USER** link and create the desired user. (In the example below, a user called ***dmf-alice***is created.)

Figure 9-39: Add a New User



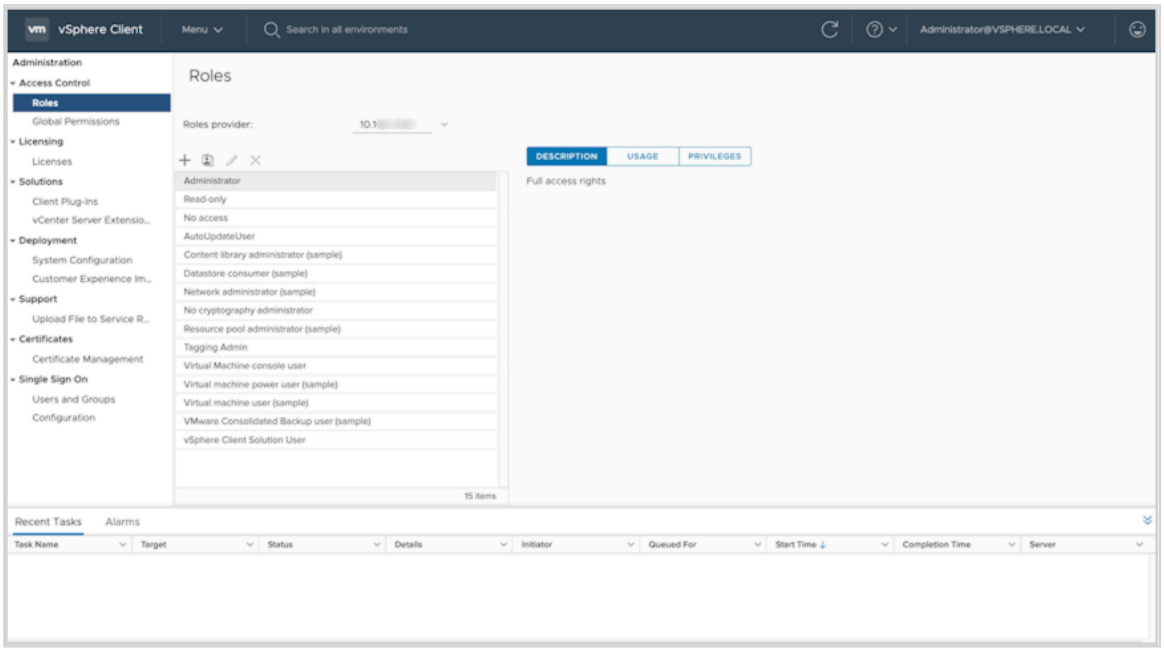
5. Verify that the newly created user is on the **Users and Groups** page.

Figure 9-40: Verify User Created



6. After creating the desired user, create and assign a role to this user. Select **Roles** under **Access Control** in the navigation bar on the left. Next, select the **+** sign to add a new role.

Figure 9-41: Add a New Role



7. In the **New Role** pop-up dialog, select **Distributed Switch** from the left and then scroll down to find and select **VSPAN operation** as the role. Select **Next** and give the new role a new name. (In the example below the new role ***monitor-dmf*** is created.) Select **Finish** to create the new role.

Figure 9-42: Select Role Type

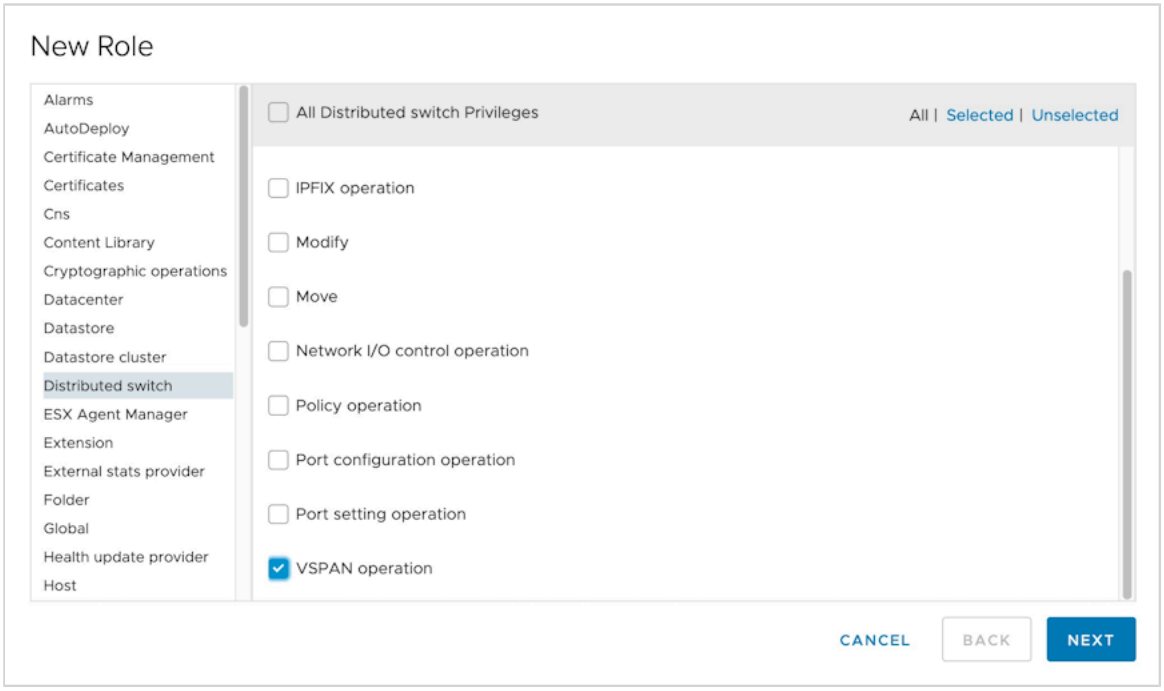
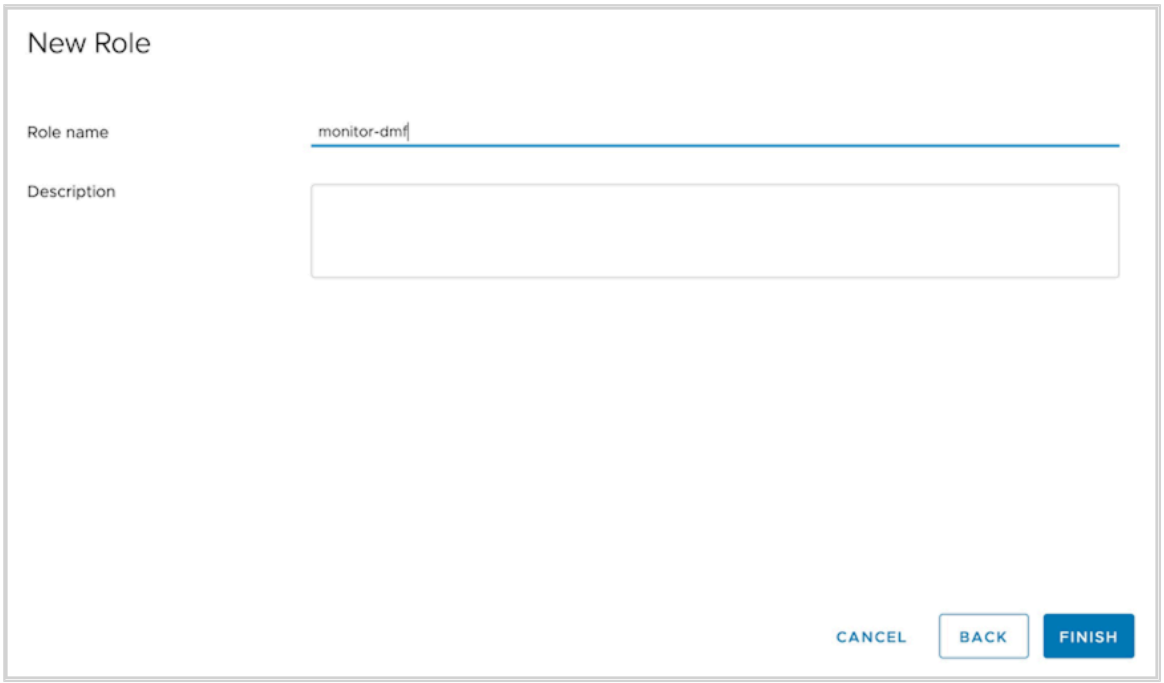
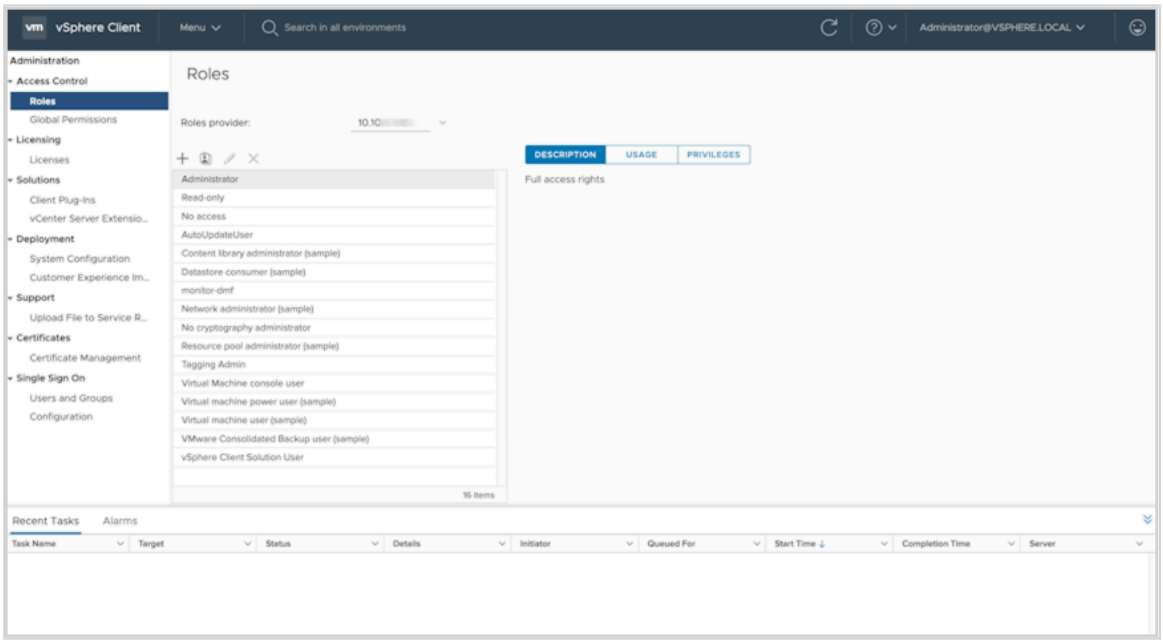


Figure 9-43: Save New Role



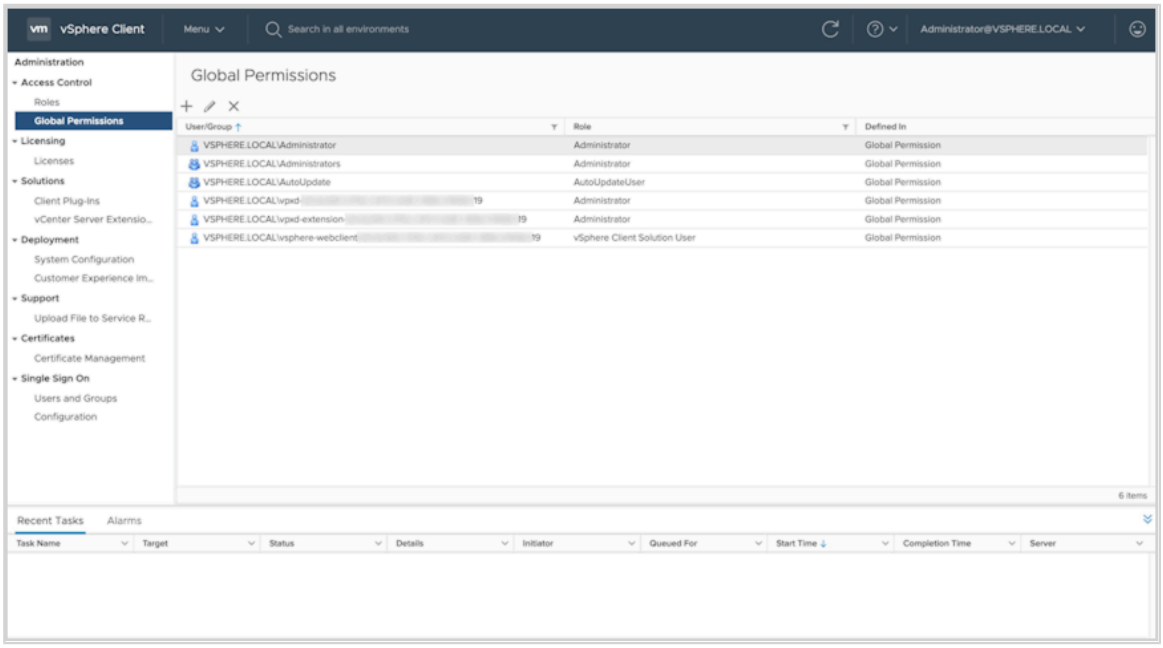
8. Verify the creation of the new role on the **Roles** page.

Figure 9-44: Verify New Role Created



9. To assign the new role to the new user, select the **Global Permissions** link in the navigation bar on the left. Next, select the **+** sign to assign the new role.

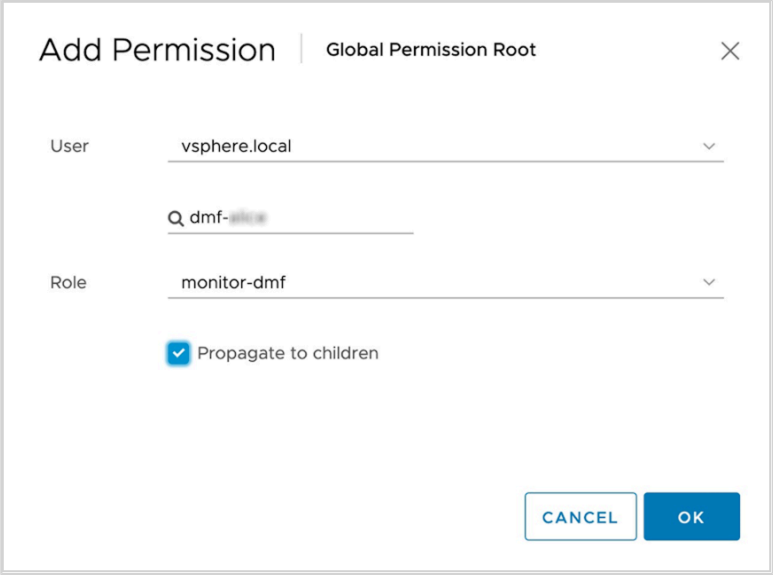
Figure 9-45: Global Permissions



10. In the **Add Permission** dialog, type the newly created username and select the newly created role, as shown in the figure below.

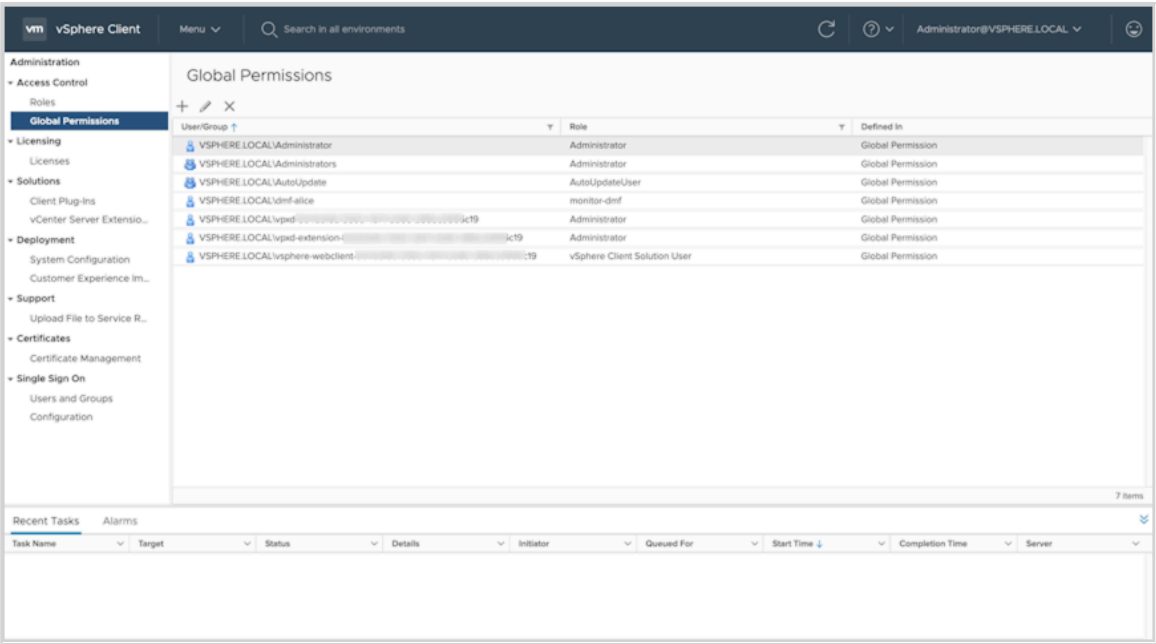
Note: Do not forget to check mark the **Propagate to children** checkbox.

Figure 9-46: Assign Role to User



11. Verify assigning the newly created role to the newly created user.

Figure 9-47: Verify Role Assignment to User



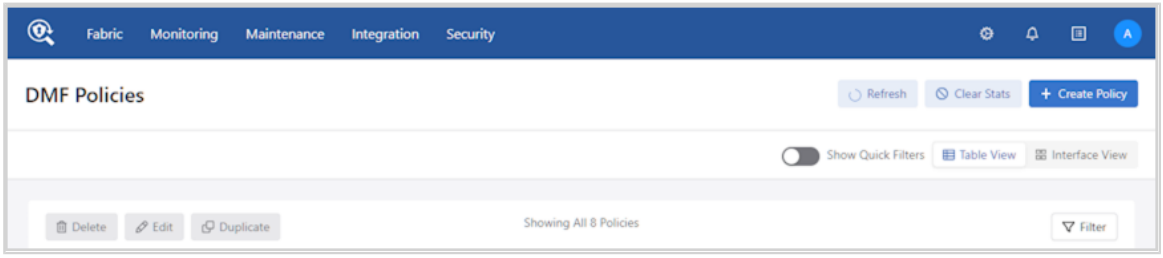
9.10 Monitor vCenter Traffic by VM Names

Match VMware vCenter-specific information in the policy. Specifically, this feature matches traffic using VMware vCenter Virtual Machine (VM) names and requires DANZ Monitoring Fabric (DMF) vCenter integration.

Configure vCenter VM name matches under the DANZ Monitoring Fabric (DMF) policies match rules section. For example:

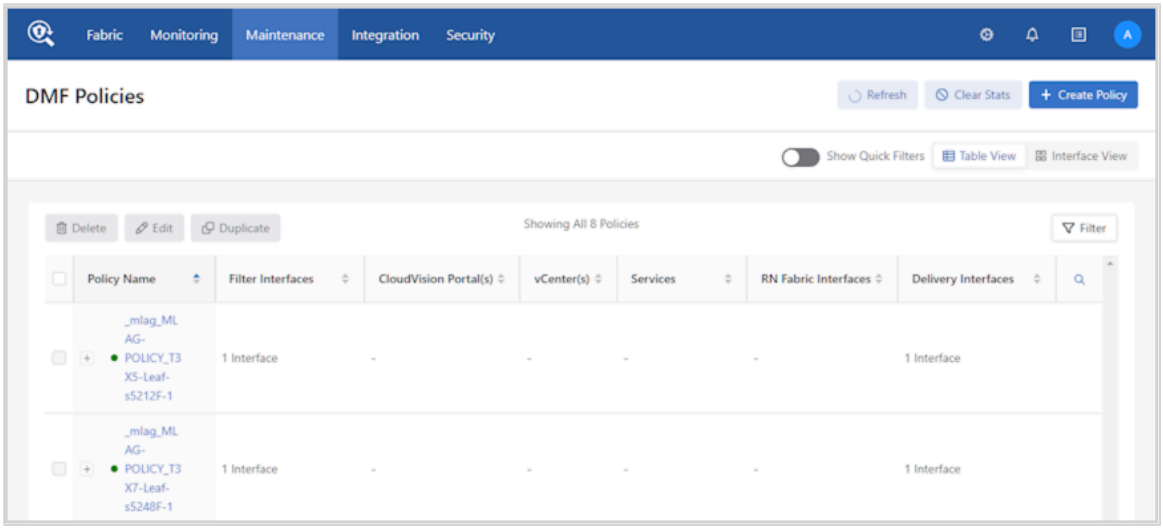
1. In the DMF GUI, navigate to the **Monitoring > Policies** page.

   Figure 9-48: DMF Policies



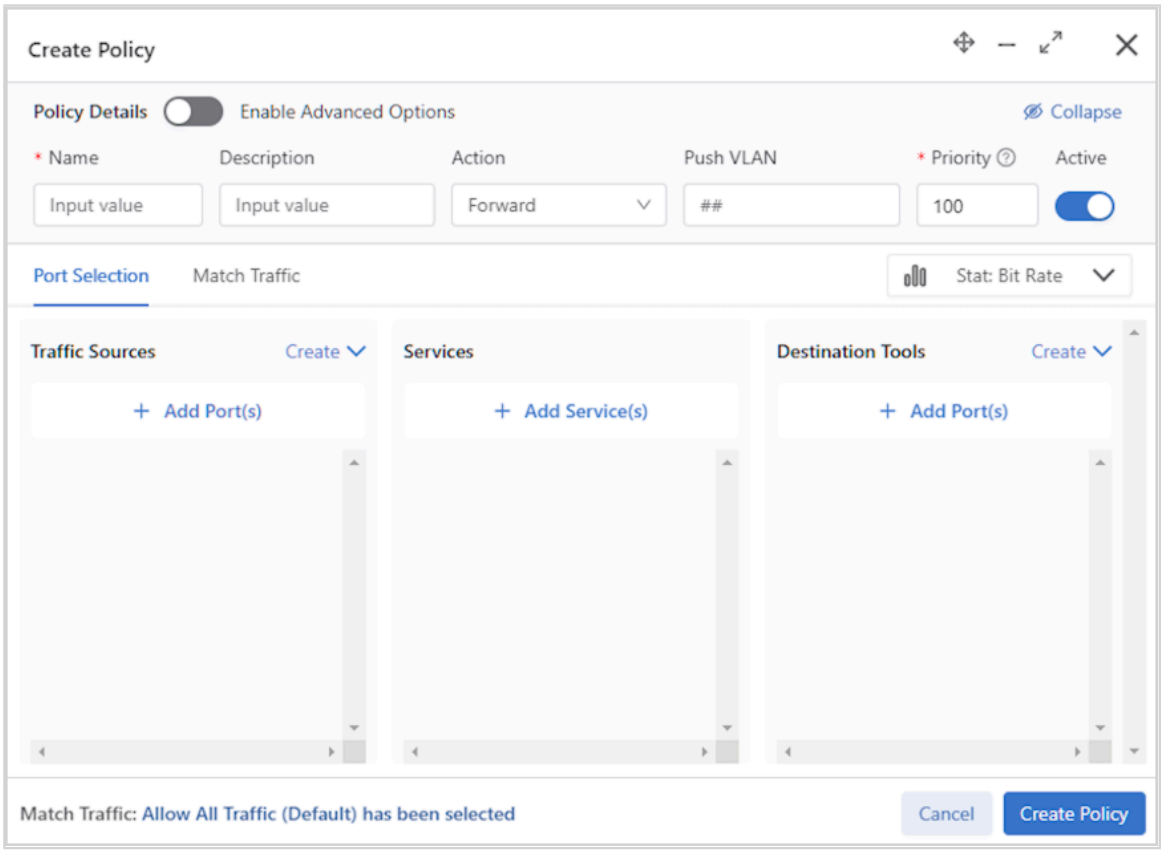
2. Select **Create Policy** to create a new policy or edit an existing one by selecting a row from the **Policies Table** and selecting **Edit**.

   Figure 9-49: Create / Edit Policy



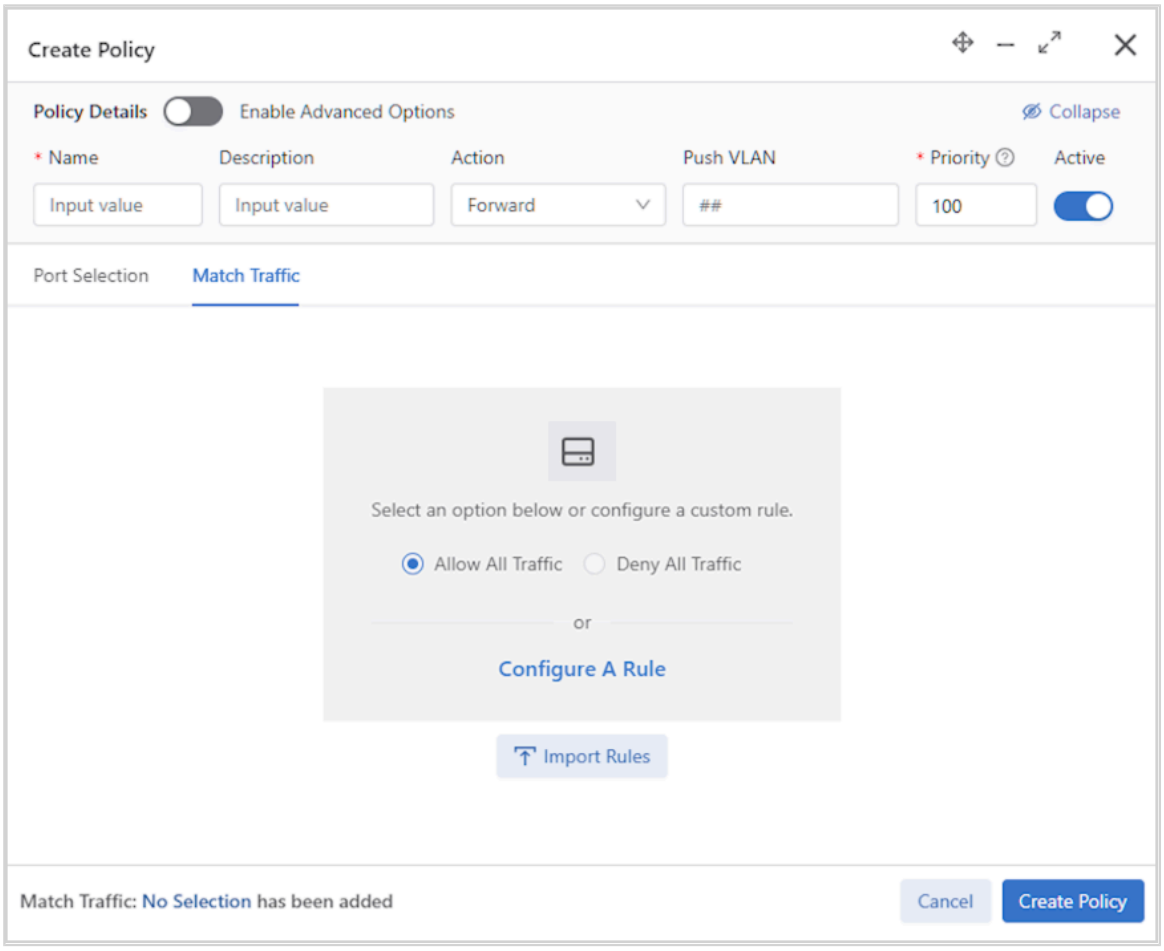
3. Navigate to the **Match Traffic** tab.

Figure 9-50: Match Traffic



4. Select **Configure a Rule** to configure a custom match rule.

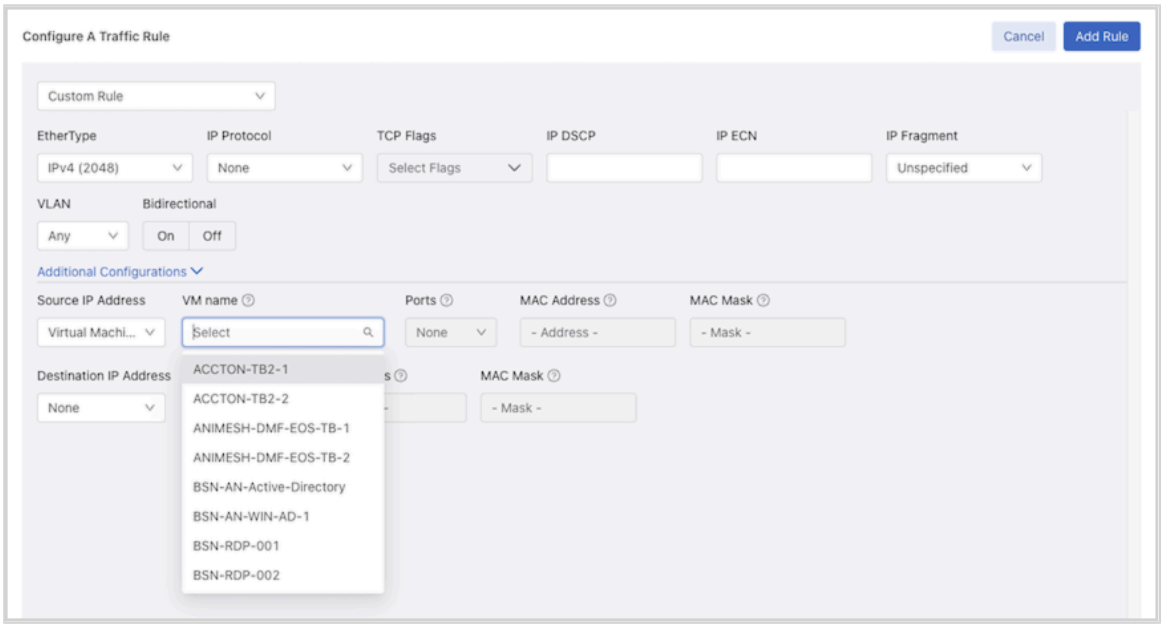
Figure 9-51: Configure a Rule



5. Set the **EtherType** to **IPv4** or **IPv6**.

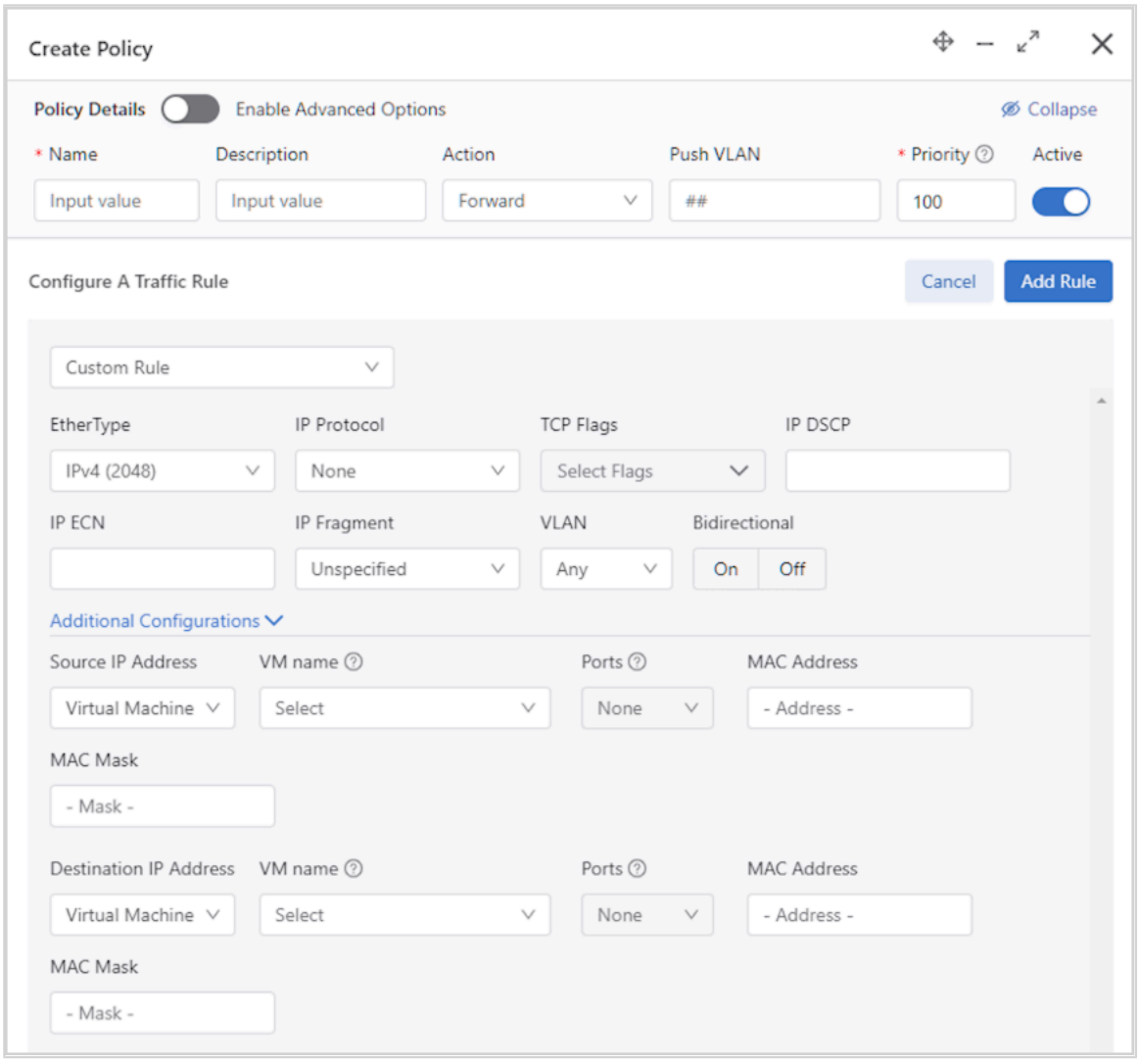
6. Add the **Source IP Address** as the vCenter VM name. Select the **Virtual Machine** option from the **Source IP Address** drop-down and select a virtual machine from the **VM Name** drop-down.

Figure 9-52: Source IP Address VM Name



7. Add the Destination IP address as the vCenter VM name. Select the **Virtual Machine** option from the **Destination IP Address** drop-down and select a virtual machine from the **VM Name** drop-down.

Figure 9-53: Destination IP VM



Note: If the **VM Name** drop-down shows **No Data**, ensure only one vCenter is affiliated with the policy (under **Traffic Sources**).

8. Select **Add Rule** to add the match rule to the policy.
9. After entering other inputs as required, select **Create Policy** (or **Save Policy**) to save the configuration.

9.10.1 Limitations

- This feature only works with vCenter integration and a direct Switch Port Analyzer (SPAN) from a switch with ESXi traffic.
- VM interface IP addresses connected to **dvs** will only be added to policy matches.
- The system may use extra TCAM entries if the management network uses **dvs**.
- **Vmkernal** names cannot be matched in the policy.
- When a VM name with multiple vNICs (multiple IP addresses) matches the policy, a TCAM entry is added for all the IP addresses.
- VM Names cannot be matched with the MAC option in the policy.
- If the vCenter becomes disconnected, policies associated with the VM names may not get correct matches or traffic.

9.11 Using the Command Line Interface

Defining a Tunnel Endpoint

Integrate a vCenter Instance

View vCenter Configuration

Integrating vCenter with DMF using Mirror Stack

Create Wildcard Tunnels

Monitor vCenter Traffic by VM Names

9.11.1 Defining a Tunnel Endpoint

To configure a tunnel endpoint using the CLI, enter the **`tunnel-endpoint`** command from config mode using the following syntax:

```
controller-1(config)# tunnel-endpoint name switch switch interface ip-address address
 mask
mask gateway address
```

For example, the following command defines ***ethernet24*** on ***F-SWITCH-1*** as a tunnel endpoint named ***OSEP1***:

```
controller-1(config)# tunnel-endpoint OSEP1 switch F-SWITCH-1 ethernet24 ip-address 172.27.1.1
mask 255.255.255.0 gateway 172.27.1.2
```

The IP address assigned to this endpoint is ***172.27.1.1***, and the next hop address for connecting to the vCenter via ***OSEP1*** (using ERSPAN) is ***172.27.1.2***.

9.11.2 Integrate a vCenter Instance

Refer to the following topics to monitor VMs using Encapsulated Remote SPAN (ERSPAN) or Switch Port Analyzer (SPAN) on a locally connected vCenter instance and VMs on a second locally connected vCenter instance.

- Monitoring VMs using ERSPAN on a Locally Connected vCenter Instance
- Monitoring VMs using SPAN on a Locally Connected vCenter Instance
- Monitoring VMs on a Second Locally Connected vCenter Instance

9.11.2.1 VMs using ERSPAN on a Locally Connected vCenter Instance

To configure the DANZ Monitoring Fabric Controller for monitoring VMs using ERSPAN on a locally connected vCenter instance, perform the following steps:

1. Add the vCenter instance details by entering the following commands.

```
controller-1(config)# vcenter vc-1
controller-1(config-vcenter)# host-name 10.8.23.70
controller-1(config-vcenter)# password 094e470e2a121e060804
controller-1(config-vcenter)# user-name root
```

2. Specify the mirror type by entering the following commands.

```
controller-1(config-vcenter)# mirror-type erspan
controller-1(config-vcenter)# sampling-rate 60
controller-1(config-vcenter)# mirrored-packet-length 60
```

   The **sampling-rate** and **mirrored-packet-length** commands are optional.

3. ERSPAN mirroring requires a tunnel endpoint configuration. Use the **cluster** command to specify a common tunnel endpoint for all the ESXi hosts in the cluster. Use the **default-tunnel-endpoint** command to specify a common tunnel endpoint for all the ESXi hosts regardless of the cluster. When using both the **cluster** and **default-tunnel-endpoint** commands, all hosts in clusters form tunnels using the cluster-specific configuration, and all the other hosts not a part of any cluster use the default configuration to form tunnels.

```
controller-1(config-vcenter)# default-tunnel-endpoint VCEP1
controller-1(config-vcenter)# cluster cluster-name tunnel-endpoint tunnel-endpoint-name
```

   Using the **tab** auto-complete feature with the cluster suggests existing cluster names associated with the vCenter.

4. Add a static route to the default or cluster tunnel-endpoint in each ESXI host.

```
esxcli network ip route ipv4 add -n network -g gateway
Example: esxcli network ip route ipv4 add -n 192.168.200.0/24  -g 192.168.150.1
```

5. Add the VMs to monitor by entering the following commands.

```
controller-1(config-vcenter)# vm-monitoring
controller-1(config-vcenter-vm-monitoring)# vm vm-2001
controller-1(config-vcenter-vm-monitoring)# vm vm-2002
```

6. Receive-only GRE tunnel-interfaces are auto-configured under **switch** for all the hosts belonging to **vc-1** that have a route to the default or cluster tunnel-endpoint.

```
! switch
switch DMF-RU34
mac 94:8e:d3:fd:6b:96
!
gre-tunnel-interface vcenter-abd08a18
direction receive-only
local-ip 192.168.200.254 mask 255.255.255.0 gateway-ip 192.168.200.1
origination vc8--interface
parent-interface ethernet55
remote-ip 192.168.150.27
gre-key-decap 33554432
!
gre-tunnel-interface vcenter-abd08a37
direction receive-only
local-ip 192.168.200.254 mask 255.255.255.0 gateway-ip 192.168.200.1
origination vc8--interface
parent-interface ethernet55
remote-ip 192.168.150.28
gre-key-decap 33554432
```

```
!
gre-tunnel-interface vcenter-abd08a56
direction receive-only
local-ip 192.168.200.254 mask 255.255.255.0 gateway-ip 192.168.200.1
origination vc8--interface
parent-interface ethernet55
remote-ip 192.168.50.29
gre-key-decap 33554432
```

7. Enter the **show running-config vcenter** command to view the vCenter configuration.

```
controller-1# show running-config vcenter
! vcenter
vcenter vc-1
hashed-password 752a3a3211040e0200090409090611
host-name 10.8.23.70
mirror-type erspan
mirrored-packet-length 60
sampling-rate 60
user-name administrator@vsphere.local
!
vm-monitoring
vm vm-2001
vm vm-2002
```

8. Configure the policies specifying the match rules and delivery interfaces.

```
controller-1(config)# policy dmf-policy-with-vcenter
controller-1(config-policy)# action forward
controller-1(config-policy)# filter-vcenter vc-1
controller-1(config-policy)# 1 match any
controller-1(config-policy)# delivery-interface TOOL-PORT-03
```

9. Enter the **show running-config policy** command to view the automatically assigned filter interfaces.

```
controller-1# show running-config policy dmf-policy-with-vcenter
! policy
policy dmf-policy-with-vcenter
action forward
delivery-interface TOOL-PORT-03
filter-interface DMF-RU34-filter-vcenter-abd08a18 vc-1--interface
filter-interface DMF-RU34-filter-vcenter-abd08a37 vc-1--interface
filter-interface DMF-RU34-filter-vcenter-abd08a56 vc-1--interface
filter-vcenter vc-1
1 match any
```

All the host tunnels belonging to **vc-1** will become the filter interfaces. If new hosts are added, deleted, or modified, policies will be recomputed with the new interfaces.

9.11.2.2 VMs using SPAN on a Locally Connected vCenter Instance

To configure the DANZ Monitoring Fabric Controller for monitoring VMs using SPAN on a locally connected vCenter instance, perform the following steps:

1. Add the vCenter instance details by entering the following commands.

```
controller-1(config)# vcenter vc-1
controller-1(config-vcenter)# host-name 10.8.23.70
controller-1(config-vcenter)# password 094e470e2a121e060804
controller-1(config-vcenter)# user-name root
```

2. Specify the mirror type by entering the following commands.

```
controller-1(config-vcenter)# mirror-type span
controller-1(config-vcenter)# sampling-rate 60
controller-1(config-vcenter)# mirrored-packet-length 60
```

   The **sampling-rate** and **mirrored-packet-length** commands are optional.

3. Add the VMs to monitor by entering the following commands.

```
controller-1(config-vcenter)# vm-monitoring
controller-1(config-vcenter-vm-monitoring)# vm vm-2001
controller-1(config-vcenter-vm-monitoring)# vm vm-2002
```

4. To view the vCenter configuration, enter the **show running-config vcenter** command as in the following example.

```
controller-1# show running-config vcenter
! vcenter
vcenter vc-1
hashed-password 752a3a3211040e0200090409090611
host-name 10.8.23.70
mirror-type span
mirrored-packet-length 60
sampling-rate 60
user-name administrator@vsphere.local
!
vm-monitoring
vm vm-2001
vm vm-2002
```

5. Configure the policies specifying the match rules and delivery interfaces.

```
controller-1(config)# policy dmf-policy-with-vcenter
controller-1(config-policy)# action forward
controller-1(config-policy)# filter-vcenter vc-1
controller-1(config-policy)# 1 match any
controller-1(config-policy)# delivery-interface TOOL-PORT-03
```

6. To view the automatically assigned filter interfaces, enter the **show running-config policy** command.

```
controller-1# show running-config policy dmf-policy-with-vcenter
! policy
policy dmf-policy-with-vcenter
action forward
delivery-interface TOOL-PORT-03
filter-interface vc-filter-1 origination vc-10-9-19-7--filter-interface
filter-interface vc-filter-3 origination vc-10-9-19-7--filter-interface
filter-vcenter vc-1
1 match any
```



Note: LLDP automatically learns the filter interfaces. All the hosts belonging to **vc-1** that have physical connections to DMF switches become the filter interfaces. If new connections are made later (or existing connections are changed), policies will be recomputed with the new interfaces.

9.11.2.3 VMs on a Second Locally Connected vCenter Instance

To configure the DMF Controller for monitoring VMs on a second locally connected vCenter instance, perform the following steps:

1. Add the VMs to monitor and configure the DMF policies to specify the match rules and delivery interfaces.

```
(config)# vcenter vc-2
(config-vcenter)# host-name 10.8.23.71
(config-vcenter)# password 094e470e2a121e060804
(config-vcenter)# user-name root
(config-vcenter)# mirror-type span | erspan
(config-vcenter)# sampling-rate 60
(config-vcenter)# mirrored-packet-length 60
(config-vcenter)# vm-monitoring
(config-vcenter-vm-monitor)# vm vm-1001
(config-vcenter-vm-monitor)# vm vm-1002
```

2. Configure the policy for the second vCenter instance.

```
(config)# policy dmf-policy-with-vcenter-2
(config-policy)# filter-vcenter vc-2
(config-policy)# 1 match any
(config-policy)# delivery-interface TOOL-PORT-02
```

9.11.3 View vCenter Configuration

To view the vCenter configuration in the CLI, use the **`show vcenter`** command, as in the following examples:

```
controller-1# show vcenter
#  vCenter Name vCenter Host Name or IP Last vCenter Update Time       Detail State                vSphere Version
--|------------|----------------------|------------------------------|---------------------------|---------------|
1  vc-10-9-0-75 10.9.0.75 2017-09-09    18:02:35.980000 PDT           Connected and authenticated. 6.5.0
2  vc-10-9-0-76 10.9.0.76 2017-09-09    18:02:36.488000 PDT           Connected and authenticated. 6.5.0
3  vc-10-9-0-77 10.9.0.77 2017-09-09    18:02:35.908000 PDT           Connected and authenticated. 6.0.0
4  vc-10-9-0-78 10.9.0.78 2017-09-09    18:02:33.507000 PDT           Connected and authenticated. 6.5.0
5  vc-10-9-0-79 10.9.0.79 2017-09-09    18:02:32.248000 PDT           Connected and authenticated. 6.5.0
6  vc-10-9-0-80 10.9.0.80 2017-09-09    18:02:32.625000 PDT           Connected and authenticated. 6.0.0
7  vc-10-9-0-81 10.9.0.81 2017-09-09    18:02:34.672000 PDT           Connected and authenticated. 6.0.0
8  vc-10-9-0-82 10.9.0.82 2017-09-09    18:02:33.008000 PDT           Connected and authenticated. 6.0.0
9  vc-10-9-0-83 10.9.0.83 2017-09-09    18:02:30.011000 PDT           Connected and authenticated. 6.0.0
10 vc-10-9-0-84 10.9.0.84 2017-09-09    18:02:33.024000 PDT           Connected and authenticated. 6.5.0
11 vc-10-9-0-85 10.9.0.85 2017-09-09    18:02:34.827000 PDT           Connected and authenticated. 6.0.0
12 vc-10-9-0-86 10.9.0.86 2017-09-09    18:02:35.164000 PDT           Connected and authenticated. 6.0.0
13 vc-10-9-0-87 10.9.0.87 2017-09-09    18:02:38.042000 PDT           Connected and authenticated. 6.5.0
14 vc-10-9-0-88 10.9.0.88 2017-09-09    18:02:37.212000 PDT           Connected and authenticated. 6.0.0
15 vc-10-9-0-89 10.9.0.89 2017-09-09    18:02:33.436000 PDT           Connected and authenticated. 6.5.0
controller-1#

controller-1# show vcenter vc-10-9-0-75
#  vCenter Name vCenter Host Name or IP Last vCenter Update Time       Detail State                vSphere Version
--|------------|----------------------|------------------------------|---------------------------|---------------|
1  vc-10-9-0-75 10.9.0.75 2017-09-09    18:02:44.698000 PDT           Connected and authenticated. 6.5.0
controller-1#

controller-1# show vcenter vc-10-9-0-75 detail
vCenter Name : vc-10-9-0-75
vCenter Host Name or IP : 10.9.0.75
Last vCenter Update Time : 2017-09-09 18:02:49.463000 PDT
Detail State : Connected and authenticated.
vSphere Version : 6.5.0
controller-1#

controller-1# show vcenter vc-10-9-0-75 error
vCenter Name : vc-10-9-0-75
vCenter Host Name or IP : 10.9.0.75
State : connected
Detail State : Connected and authenticated.
Detailed Error Info :
controller-1#
```

9.11.4 Integrating vCenter with DMF using Mirror Stack

From the DMF Controller configure the TCP/IP stack using the **`tcp-ip-stack`** option in the vCenter config. The default and recommended value is **`mirror-stack`**.

```
dmf-controller-1(conf)# vcenter vc8
dmf-controller-1(config-vcenter)# tcp-ip-stack
default-stack mirror-stack
dmf-controller-1(config-vcenter)# tcp-ip-stack mirror-stack
```

Show Commands

Use the **`show running-config`** command to view the **`tcp-ip-stack`** configuration.



Note: If **`mirror-stack`** is configured, it will only show when using the **`details`** token.

```
dmf-controller-1(config-vcenter)# show running-config vcenter vc8 details

! vcenter
vcenter vc8
default-tunnel-endpoint r34-lag-leaf1b
hashed-password <hashed-password>
host-name <ip-address>
mirror-type encapsulated-remote
tcp-ip-stack mirror-stack
user-name administrator@vsphere.local
```

View the existing mirror stack NICs and IPs of the host using the **`show vcenter vCenter name inventory`** command.



Note: **`v8`** is an example vCenter name.

```
dmf-controller-1# show vcenter vc8 inventory
# vCenter ESXi Host     Host DNS Name                        Cluster         Product Name
          Hardware Model CPU Usage (%) Memory Usage (%) Virtual switches Mirror Stack VMkernel
 Adapter VMkernel Adapter IP Address
-|-------|-------------|-----------------------------------|---------------|--------------------
------------|--------------|-------------|----------------|----------------|---------------
--------------|---------------------------|
1 vc8     10.240.166.27 ESX27.qa.bsn.sjc.aristanetworks.com BSN-NSX-1       VMware ESXi 8.0.2
 build-22380479 PowerEdge R430 2             15               3                vmk1
           192.168.60.27
2 vc8     10.240.166.28 ESX28.qa.bsn.sjc.aristanetworks.com BSN-NSX-2       VMware ESXi 8.0.2
 build-22380479                0             4                4
3 vc8     10.240.166.29 ESX29.qa.bsn.sjc.aristanetworks.com Edge            VMware ESXi 8.0.0
 build-20513097 PowerEdge R430 4             23               3
4 vc8     10.240.166.33 ESX33.qa.bsn.sjc.aristanetworks.com vc8-mixed-stack VMware ESXi 8.0.2
 build-22380479                0             6                3                vmk1
           192.168.60.33
5 vc8     10.240.166.35 ESX35.qa.bsn.sjc.aristanetworks.com MGMT            VMware ESXi 7.0.2
 build-17867351 PowerEdge R430 26            23               2
6 vc8     10.240.166.38 ESX38.qa.bsn.sjc.aristanetworks.com vc8-mixed-stack VMware ESXi 8.0.2
 build-22380479                1             23               3                vmk1
           192.168.60.38
dmf-rack#
```

9.11.4.1 Troubleshooting

Use the **show fabric errors** and **show fabric warnings** commands to troubleshoot and verify that everything is functioning as expected.

In the following example, the error message indicates that DMF could not find a route from the ESXi host to the DMF tunnel endpoint.

```
dmf-controller-1# show fabric errors
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ vCenter related error
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#  vCenter Name Error
--|------------|-----------------------------------------------------------------------------------------
-----------------------------------------------------------|
1  vc701        Unable to locate a matching route for Mirror TCP/IP stack in host ESX37.qa.bsn.
sjc.aristanetworks.com for DMF endpoint 192.168.200.254
```

9.11.5 Create Wildcard Tunnels

The current implementation of VMware vCenter creates one tunnel interface from every ESXi host to DMF.

Using a wildcard tunnel on DMF for VMware vCenter reduces the number of tunnels created.

Platform Compatibility

This feature is only compatible with switches that support wildcard tunneling.

The CLI construct **`wildcard-tunnels`** is available as a configuration option when configuring a VMware vCenter in DANZ Monitoring Fabric (DMF), as shown below:

Table 6: Commands

`cluster`	Configure tunnel-endpoint for cluster
`default-tunnel-endpoint`	Configure tunnel endpoints
`description`	Describe this vCenter
`hashed-password`	Set the vCenter password (to log into vCenter)
`host-name`	Set the vCenter hostname
`mirror-type`	Set the vCenter vm monitoring mode
`mirrored-packet-length`	Set the mirrored packet length
`password`	Set the vCenter password (to log into vCenter)
`sampling-rate`	Set the packet sampling rate
`user-name`	Set the vCenter user name (to log into vCenter)
`vm-monitoring`	Enter `vm-monitoring config` submode
`wildcard-tunnels`	Enable wildcard tunnels

Enable wildcard tunnels by setting the above leaf parameter, as shown in the following example of vCenter configuration on the Controller node.

```
dmf-controller-1(config)# vcenter VC1
dmf-controller-1(config-vcenter)# wildcard-tunnels
dmf-controller-1(config-vcenter)# show this
! vcenter
vcenter VC1
wildcard-tunnels
dmf-controller-1(config-vcenter)#
```

Similarly, disable wildcard tunnels by issuing the **no** command as shown below:

```
dmf-controller-1(config-vcenter)# show this
! vcenter
vcenter VC1
wildcard-tunnels
dmf-controller-1(config-vcenter)# no wildcard-tunnels
dmf-controller-1(config-vcenter)# show this
! vcenter
vcenter VC1
dmf-controller-1(config-vcenter)#
```

Show Commands

There is no specific show command for wildcard tunnels; however, check them in the vCenter running config. In addition, the **show tunnels** command shows the tunnels created for the selected vCenter configuration with a wildcard remote IP address.

Troubleshooting

Verify errors and warnings are clear using the **show fabric errors** and **show fabric warnings** commands. The **show tunnels** command displays tunnels created based on the vCenter configuration on the Controller with a wildcard remote IP address. Use the **show switch *name* table gre-tunnel** command to display tunnels programmed on the switch.

9.11.6 Monitor vCenter Traffic by VM Names

Configuration

This feature works with vCenter integration; therefore, configure vCenter Integration in DANZ Monitoring Fabric (DMF). Configure vCenter mapping in the policy, then define a policy match using VM names in the vCenter as illustrated in the following configuration example:

```
dmf-controller-1(config)# policy v1
dmf-controller-1(config-policy)# action forward
dmf-controller-1(config-policy)# filter-interface filter-interface
dmf-controller-1(config-policy)# delivery-interface delivery-interface
dmf-controller-1(config-policy)# filter-vcenter vcenter-name
dmf-controller-1(config-policy)# 1 match ip src-vm-name vm-name dst-vm-name vm-name
dmf-controller-1(config-policy)# 2 match ip6 src-vm-name vm-name
```

Show Commands

Enter the **show running-config policy *policy name*** command to display the configuration.

```
dmf-controller-1# show running-config policy v1

! policy
policy v1
action forward
delivery-interface delivery-interface
filter-interface filter-interface
filter-vcenter vcenter-name
1 match ip src-vm-name vm-name dst-vm-name vm-name
2 match ip6 src-vm-name vm-name
```

The **show policy *policy name*** command displays the policy information, including stats.

```
dmf-controller-1# show policy v2
Policy Name        : v2
Config Status      : active - forward
Runtime Status     : installed
Detailed Status    : installed - installed to forward
Priority           : 100
Overlap Priority   : 0
# of switches with filter interfaces   : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces  : 0
# of filter interfaces                 : 1
# of delivery interfaces               : 1
# of core interfaces                   : 0
# of services      : 0
# of pre service interfaces            : 0
# of post service interfaces           : 0
Push VLAN          : 5
Post Match Filter Traffic              : -
Total Delivery Rate: -
Total Pre Service Rate                 : -
Total Post Service Rate                : -
Overlapping Policies                   : none
Component Policies : none
Installed Time     : 2023-12-21 19:00:39 UTC
Installed Duration : 50 minutes, 11 secs
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Match Rules ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Rule
-|-------------------------------------------------------------------------|
1 1 match ip src-vm-name DMF-RADIUS-SERVER-1 dst-vm-name DMF-TACACS-SERVER-1

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~
# DMF IF           Switch      IF Name    State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset
 Time
-|----------------|-----------|----------|-----|---|-------|-----|--------|--------|--
----------------------------|
1 span_from_arista Arista-7050 ethernet20 up    rx  0       0     0        -        2023-12-21
 19:00:39.941000 UTC

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~
# DMF IF       Switch      IF Name      State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset
 Time
-|------------|-----------|------------|-----|---|-------|-----|--------|--------|----
--------------------------|
1 ubuntu-tools Arista-7050 ethernet49/2 up    tx  0       0     0        -        2023-12-21
 19:00:39.941000 UTC
~ Service Interface(s) ~
None.
~ Core Interface(s) ~
None.
~ Failed Path(s) ~
None.
```

The **show vcenter *vcenter name* endpoint** command displays the vCenter VM information, including networks.

```
dmf-controller-1# show vcenter vcenter1 endpoint
#  vCenter Name VM Name   ESXi Host Name Network Interface Name MAC Address                IP
 Address                                 Virtual Switch Portgroup     Power State
--|------------|---------|--------------|---------------------|--------------------------
|-----------------------------------------|--------------|-------------|-----------|
1  vcenter1     ub-11-216 10.240.155.216 Network adapter 1      00:50:56:8b:4d:03 (VMware)
 1.1.11.216/24, fe80::250:56ff:fe8b:4d03/64 DVS-DMF        vlan11        powered-on
2  vcenter1     ub-12-216 10.240.155.216 Network adapter 1      00:50:56:8b:72:a0 (VMware)
 1.1.12.216/24, fe80::250:56ff:fe8b:72a0/64 DVS-DMF        vlan12        powered-on
3  vcenter1     ub-13-216 10.240.155.216 Network adapter 1      00:50:56:8b:c0:06 (VMware)
 1.1.13.216/24, fe80::250:56ff:fe8b:c006/64 DVS-DMF        vlan-10       powered-on
4  vcenter1     ub-14-216 10.240.155.216 Network adapter 1      00:50:56:8b:d1:d9 (VMware)
 1.1.14.216/24, fe80::250:56ff:fe8b:d1d9/64 DVS-DMF        vlan-10       powered-on
```

9.11.6.1 Troubleshooting

Fabric errors and warnings are very useful for troubleshooting this feature.

When using the **`show fabric warnings`** command, the following validation message displays when the vCenter integration cannot resolve the IP address for the VM name used in the policy.

```
dmf-controller-1# show fabric warnings
~~~
```

Chapter 10

Integrating VMware NSX with DMF

This chapter describes integrating VMware NSX with the DANZ Monitoring Fabric (DMF).

- Overview
- NSX Compatibility and Requirements
- Instance Configuration
- Monitoring Configurations in Policies
- Show Commands
- Troubleshooting
- Considerations
- Resources

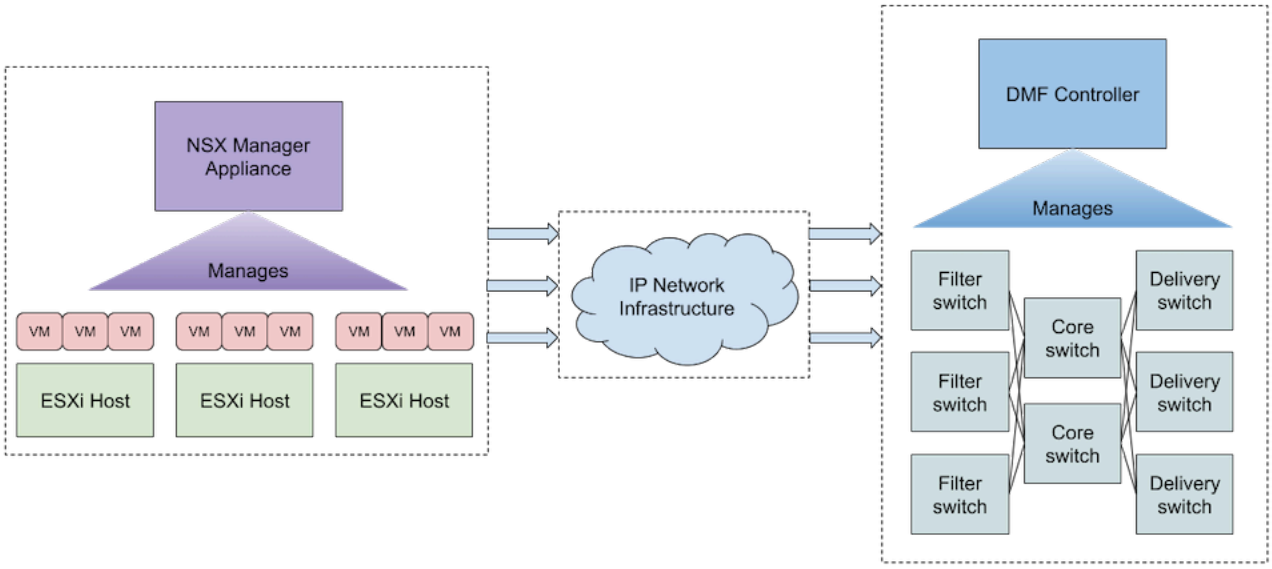
10.1 Overview

The DANZ Monitoring Fabric (DMF) allows the integration and monitoring of virtual machines in a VMware NSX fabric deployed in a vSphere environment. The DMF Controller communicates with NSX to retrieve its managed inventory and configures port mirroring sessions for selected virtual machines managed by the NSX fabric.

The policy configuration on the DMF Controller specifies which VMs to monitor and the direction of mirrored traffic (bidirectional, ingress, or egress). NSX Remote L3 SPAN mirror sessions send monitoring traffic via GRE tunneling to a Tunnel Endpoint (TEP) configured in DMF.

The following figure illustrates a simple DMF-NSX integration configuration where DMF and NSX communicate. DMF monitors traffic from NSX to a TEP configured on the fabric using an L2GRE tunnel via the IP network between the NSX and DMF fabrics.

Figure 10-1: DMF - NSX Integration



10.2 NSX Compatibility and Requirements

NSX Compatibility

NSX version 4.1 and later is supported.

NSX Requirements

The user configured in DMF for NSX integration must have sufficient permissions in NSX. The minimum permissions required are:

- **Networking:** read-only access for Connectivity.
- **Inventory:** full access for Groups and read-only access for Virtual Machines.
- **Troubleshooting:** full access for Port Mirroring.

The user configured in DMF for the vCenter compute manager must have at least read-only permissions in vCenter.

Monitoring a given VM requires:

- Installing VMware Tools on the guest operating system, thus enabling IP address discovery for the VM network adapters.
- Configuring the vCenter compute manager that manages the VM under the NSX instance configuration in DMF.
- The host transport node where the VM resides has a mirror stack VMkernel adapter configured in vCenter. Refer to the Integrating vCenter with DMF using Mirror Stack - vCenter Configuration section for details on configuring the mirror stack on an ESXi host.
- At least one VM network adapter is attached to an NSX logical port.



Note: When upgrading from DMF 8.7 with an existing NSX policy configuration to DMF 8.8 or later, Arista recommends removing this configuration before the upgrade and re-adding it after the upgrade is complete.

10.3 Instance Configuration

To integrate with NSX, configure an NSX instance in the DMF Controller, enabling communication between NSX and DMF. The NSX hostname or IP address must be reachable from the DMF Controller. If the NSX Manager is a cluster (typically three nodes on separate hypervisor hosts in a production deployment), using a hostname that will resolve to the active node is recommended to maintain the connection in case of a cluster member failure. The user in the NSX integration configuration must have at least the permissions in NSX as outlined in NSX Requirements.

VMware NSX Integration using the CLI

VMware NSX Integration using the GUI

10.3.1 VMware NSX Integration using the CLI

Configure an NSX instance in the DMF Controller:

```
dmf-controller(config)# nsx instance_name
dmf-controller(config-nsx)# host-name hostname_or_ip
dmf-controller(config-nsx)# username username
dmf-controller(config-nsx)# password password
```

Optionally, add a description to the NSX instance:

```
dmf-controller(config-nsx)# description description_of_instance
```

This configuration is sufficient for some visibility features of the NSX integration. However, vCenter compute manager(s) must also be configured for the NSX instance to allow for monitoring of managed VMs:

```
dmf-controller(config-nsx)# vcenter-compute-manager vcenter_instance_name
dmf-controller(config-nsx-vcenter-compute-manager)# host-name vcenter_hostname_or_ip
dmf-controller(config-nsx-vcenter-compute-manager)# username username
dmf-controller(config-nsx-vcenter-compute-manager)# password password
```

To refresh the connection between DMF and NSX, use the **`sync nsx`** command, which sends a request to NSX to re-authenticate the connection and to re-fetch the inventory:

```
dmf-controller# sync nsx instance_name
```

To use L2GRE tunnels in the integration, enable tunneling in the DMF Controller. For SWL switches, set the match mode to one of the following that is compatible with tunneling: **`full-match`** or **`l3-l4-offset-match`**. For EOS switches, tunneling will work in any match mode.

```
dmf-controller(config)# tunneling
Tunneling is an Arista Licensed feature.
Please ensure that you have purchased the license for tunneling before using this feature.
Enter "yes" (or "y") to continue: y
dmf-controller(config)# match-mode tunneling_compatible_mode
```

Configure tunnel endpoints to allow monitoring from NSX to DMF using an L2GRE tunnel; add a tunnel endpoint to a policy configuration or, optionally, to an NSX integration instance's configuration to define a default tunnel endpoint for this instance.

To configure a tunnel endpoint for EOS switches, use the following command:

```
dmf-controller(config)# tunnel-endpoint tep_name switch fabric_switch fabric_switch_interface ip-
address tep_ip mask subnet_mask gateway gateway_ip
```

For SWL switches, the mask and gateway parameters are optional in the tunnel-endpoint command:

```
dmf-controller(config)# tunnel-endpoint tep_name switch fabric_switch fabric_switch_interface ip-
address tep_ip
```

To set a default tunnel endpoint for an NSX integration instance, use the following commands:

```
dmf-controller(config)# nsx instance_name
dmf-controller(config-nsx)# default-tunnel-endpoint tep_name
```

To remove a default tunnel endpoint for an NSX integration instance, use the following commands:

```
dmf-controller(config)# nsx instance_name
dmf-controller(config-nsx)# no default-tunnel-endpoint tep_name
```

The NSX instance configuration flag **preserve-mirror-sessions** indicates whether mirroring sessions are preserved for the NSX instance when uninstalling DMF policies configured with it. By default, the flag is **false**, meaning existing mirroring sessions will automatically be removed if the relevant DMF policies are uninstalled.



Note: Removing the NSX monitoring config from a policy will remove the mirroring sessions regardless of the flag's setting.

Enable preserving mirroring sessions:

```
dmf-controller(config-nsx)# preserve-mirror-sessions
```

Disable preserving mirroring sessions (default behavior):

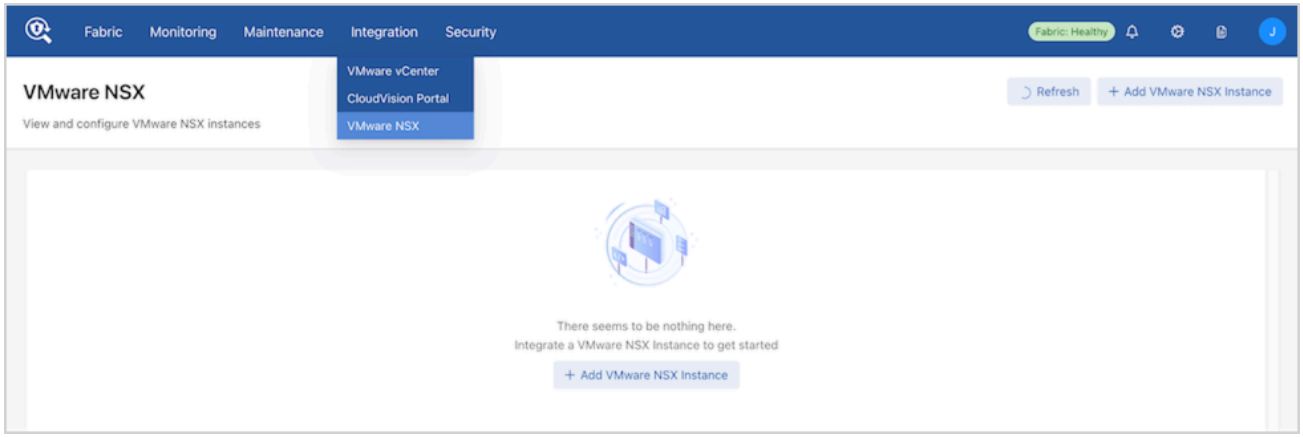
```
dmf-controller(config-nsx)# no preserve-mirror-sessions
```

10.3.2 VMware NSX Integration using the GUI

Navigate to **Integration** > **VMware NSX**.

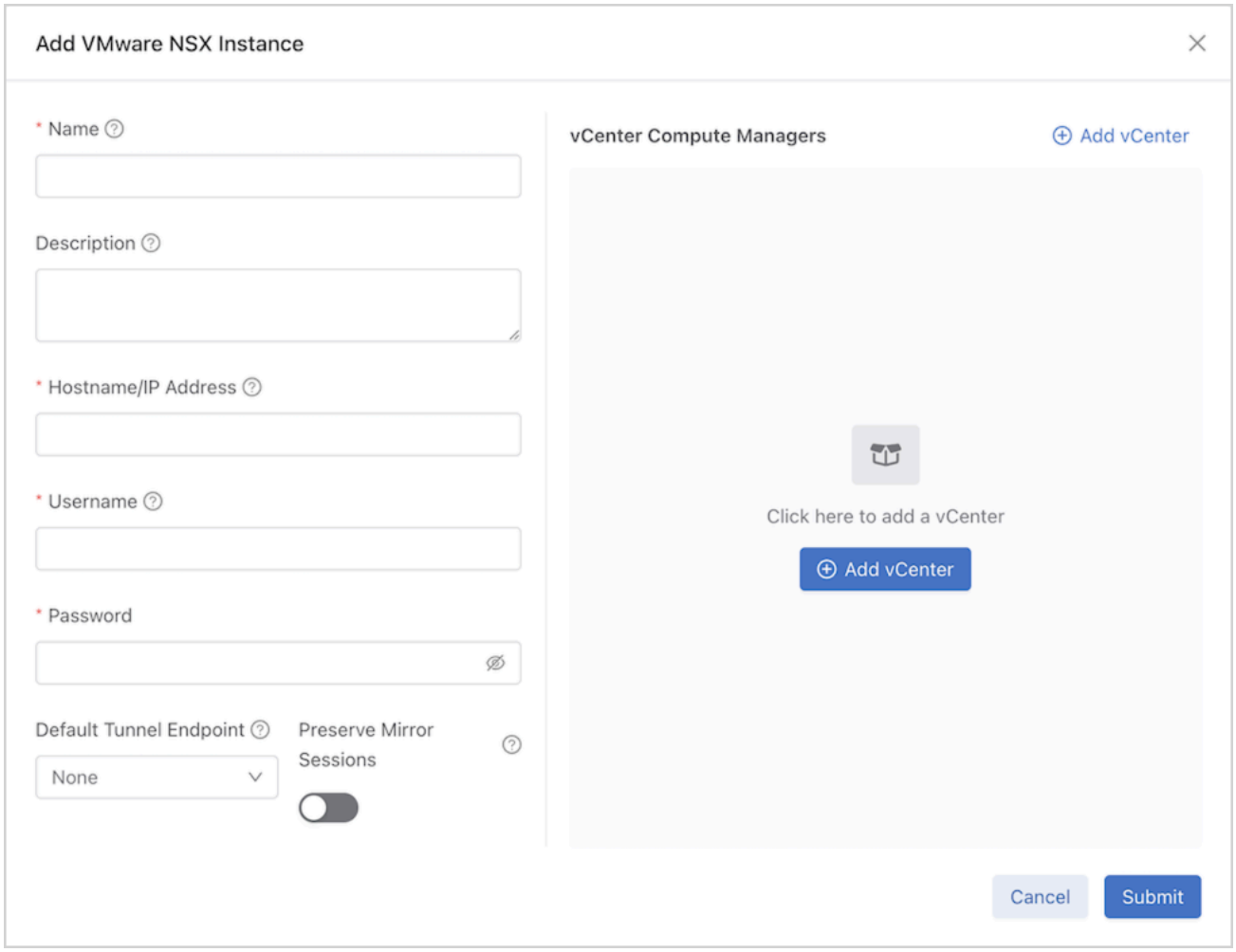
Select **Add VMware NSX Instance** to create a new NSX integration instance.

Figure 10-2: VMware NSX



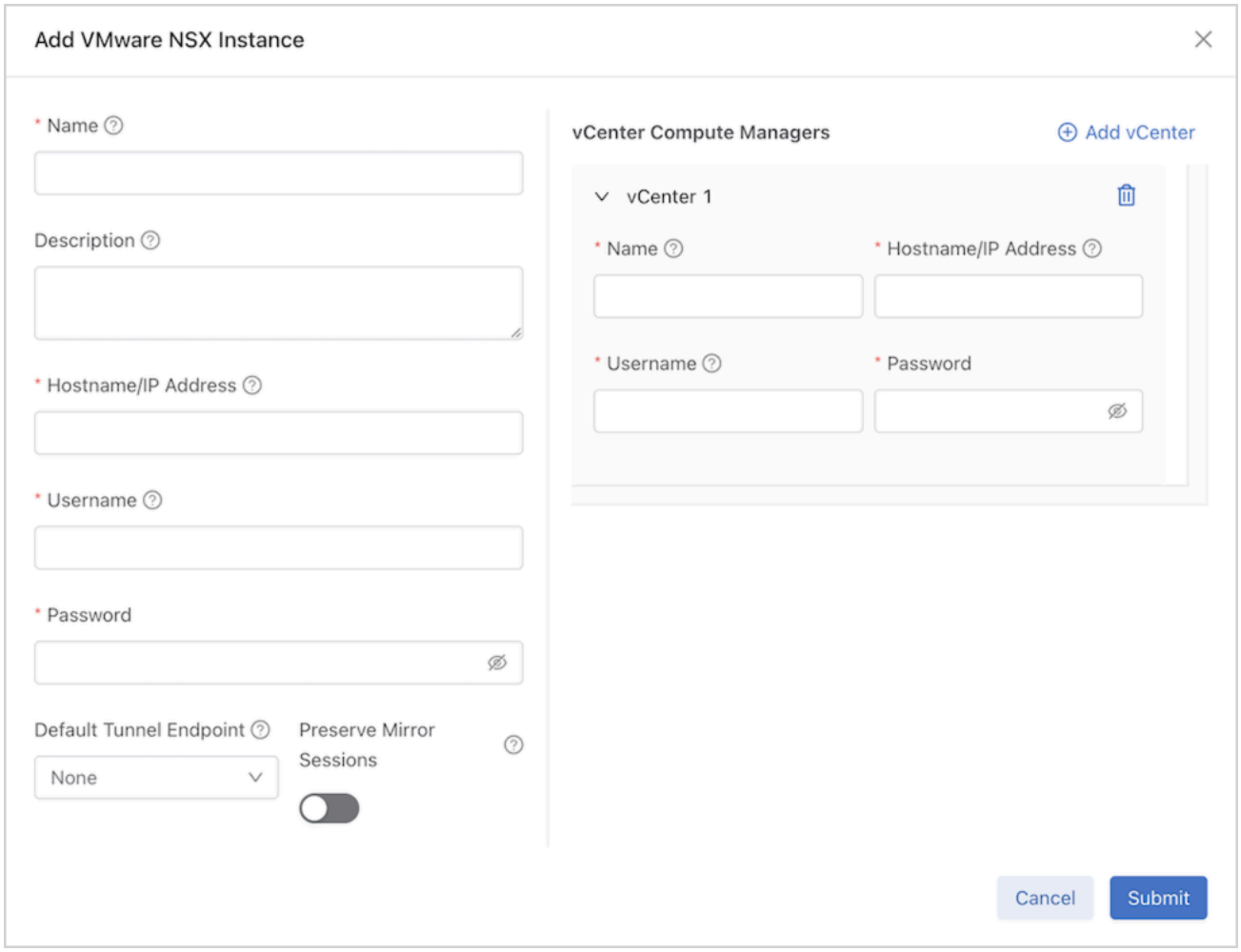
Enter the NSX integration configuration details.

Figure 10-3: Add VMware NSX Instance



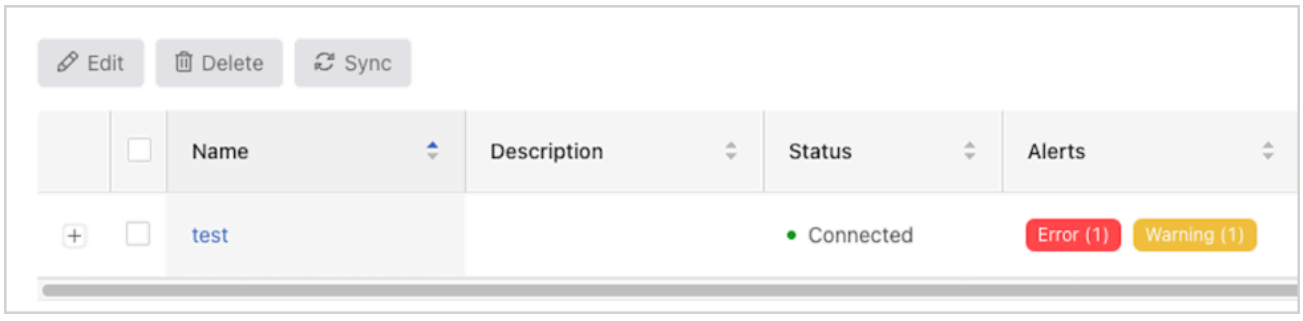
Select **Add vCenter** to add further configuration details for the vCenter compute manager(s) associated with the NSX Manager.

Figure 10-4: Add vCenter Compute Managers



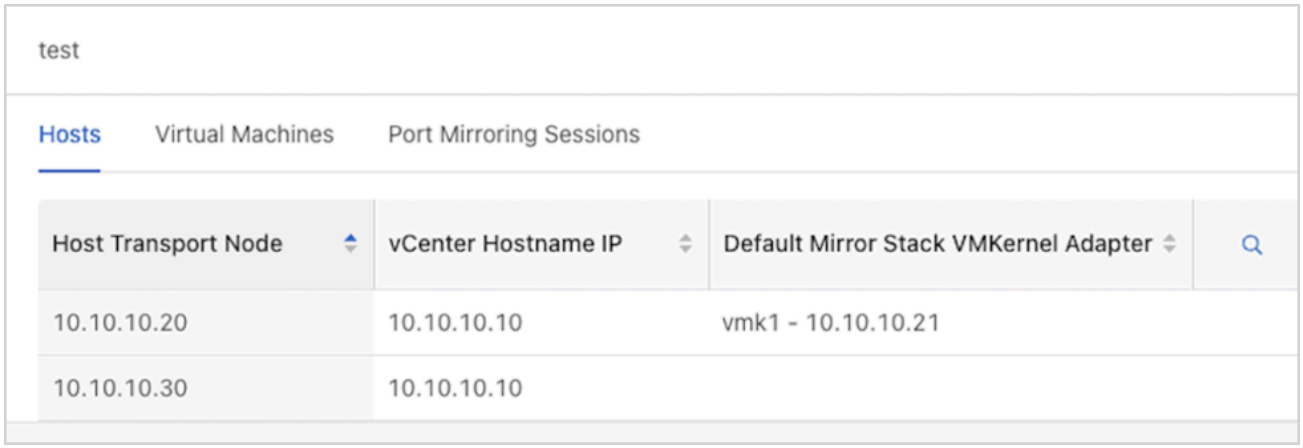
If there are any warnings or errors with the NSX integration instance, select the **tag** in the **Alerts** column to see more details.

Figure 10-5: Alerts



Select the NSX instance under **Name** to view its inventory details, including the hosts, virtual machines, and port mirroring sessions.

Figure 10-6: Inventory Details



10.4 Monitoring Configurations in Policies

DMF uses policies to create, update, or remove the monitoring of NSX virtual machines. Currently, NSX port mirroring using Remote L3 SPAN with GRE is supported.

After enabling NSX integration in a DMF policy (e.g., adding an instance as a traffic source and specifying VMs to monitor), the DMF Controller will automatically create filter and tunnel interfaces, with origination **`auto-generated`**. A mirroring session is automatically created in NSX.

Monitoring Configurations in Policies using the CLI

Monitoring Configurations in Policies using the GUI

10.4.1 Monitoring Configurations in Policies using the CLI

Add an NSX integration instance as a traffic source in a DMF policy:

```
dmf-controller(config)# policy policy_name
                dmf-controller(config-policy)# filter-nsx instance_name
                dmf-controller(config-policy-nsx)#
```

To monitor VM traffic, choose from two options:

1. Configure a tunnel endpoint as the destination for a VM's traffic in a DMF policy.
2. Configure a default tunnel endpoint destination for an NSX instance and omit the destination for a VM in a DMF policy.

Both options support an optional mirrored traffic direction. Configure this direction to:

1. bidirectional (default when omitted)
2. ingress only
3. egress only.



Note: The DMF mirrored traffic direction is from the VM's perspective (i.e., ingress traffic is entering the VM, egress traffic is exiting the VM), whereas the NSX mirror session direction is from the vSphere Distributed Switch's perspective (i.e., ingress traffic is entering the VDS from a VM, and egress traffic is exiting the VDS from a VM). Thus, it is expected that the session direction on the NSX Manager is reversed from the VM direction in the DMF policy for unidirectional mirroring.

Configure the NSX source VMs to mirror traffic to a tunnel endpoint configured in DMF, with an optional direction:

```
dmf-controller(config-policy-nsx)# vm-name vm_name gre-tunnel-endpoint tep_name
 [ direction direction ]
```

Configure the NSX source VMs to mirror traffic to the default tunnel endpoint specified in the NSX instance's configuration, with an optional direction:

```
dmf-controller(config-policy-nsx)# vm-name vm_name [ direction direction ]
```

Example

The following is an example of two DMF policies' configuration, where **testPolicy1** is monitoring bidirectional traffic from VM **vm1** in the NSX instance **test**, to the default tunnel endpoint for the instance, **TEP1**, and forwarding the traffic to the delivery interface **tool1**; **testPolicy2** is monitoring egress traffic from VM **vm2** in the same NSX instance **test**, to the tunnel endpoint **TEP2**, and forwarding the traffic to the delivery interface **tool2**. The vCenter compute manager **vcm** manages both VMs.

```
! nsx
                nsx test
                default-tunnel-endpoint TEP1
                hashed-password abc123
                host-name test.arista.com
                user-name admin
                vcenter-compute-manager vcm
                hashed-password abc123
                host-name 10.10.10.10
                user-name administrator@vsphere.local

                ! tunnel-endpoint
                tunnel-endpoint TEP1 switch leaf1 ethernet1 ip-address 10.10.10.1
                tunnel-endpoint TEP2 switch leaf2 ethernet1 ip-address 10.10.10.2

                ! policy
                policy testPolicy1
                action forward
                delivery-interface tool1
                1 match any
                filter-nsx test
                !
                vm-name vm1

                policy testPolicy2
                action forward
                delivery-interface tool2
                1 match any
                filter-nsx test
                !
                vm-name vm2 gre-tunnel-endpoint TEP2 direction egress
```



Note: If the TEP destination and traffic mirroring direction are identical, the same source VMs can be monitored across multiple DMF policies. These policies will reuse the same filter and tunnel interfaces and NSX mirror sessions. IP address match rules are automatically generated for each DMF policy based on the NSX VM network adapter IPs, to ensure that each policy receives only the mirrored traffic for the VMs it has configured.

To stop monitoring a VM on NSX, remove its configuration from the DMF policy using the following command:

```
dmf-controller(config-policy-nsx)# no vm-name vm_name
```

Stop monitoring an NSX instance in a DMF policy and all its VMs:

```
dmf-controller(config)# policy policy_name
                dmf-controller(config-policy)# no filter-nsx instance_name
```

To disable integration with an NSX instance, remove its NSX integration instance configuration and remove it from all DMF policies using the following commands:

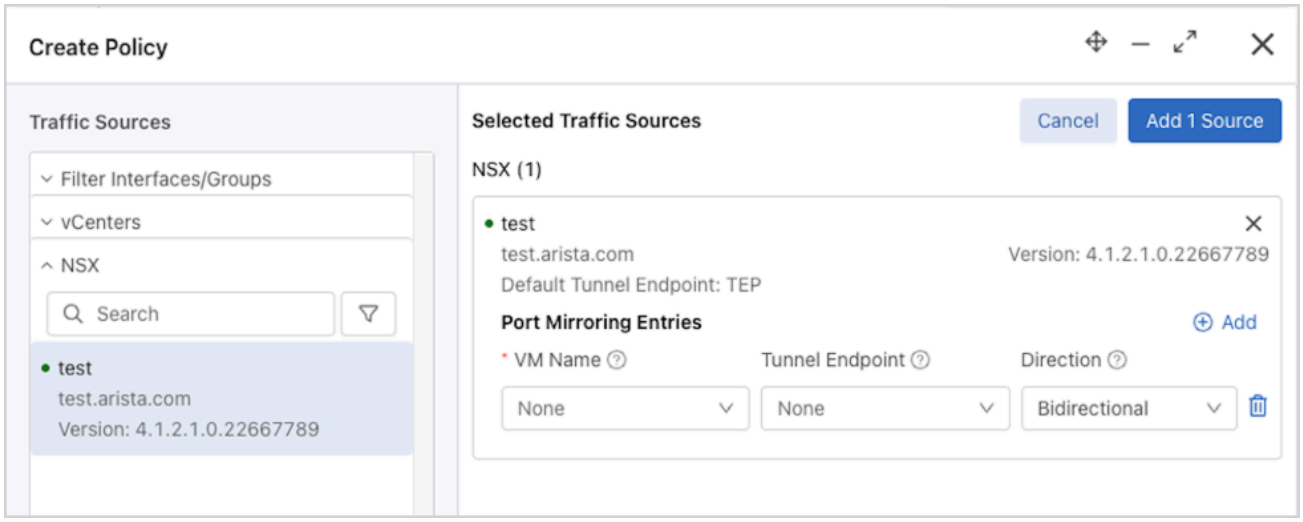
```
dmf-controller(config)# no nsx instance_name
                dmf-controller(config)# policy policy_name
                dmf-controller(config-policy)# no filter-nsx instance_name
```

10.4.2 Monitoring Configurations in Policies using the GUI

To start monitoring traffic from the production network:

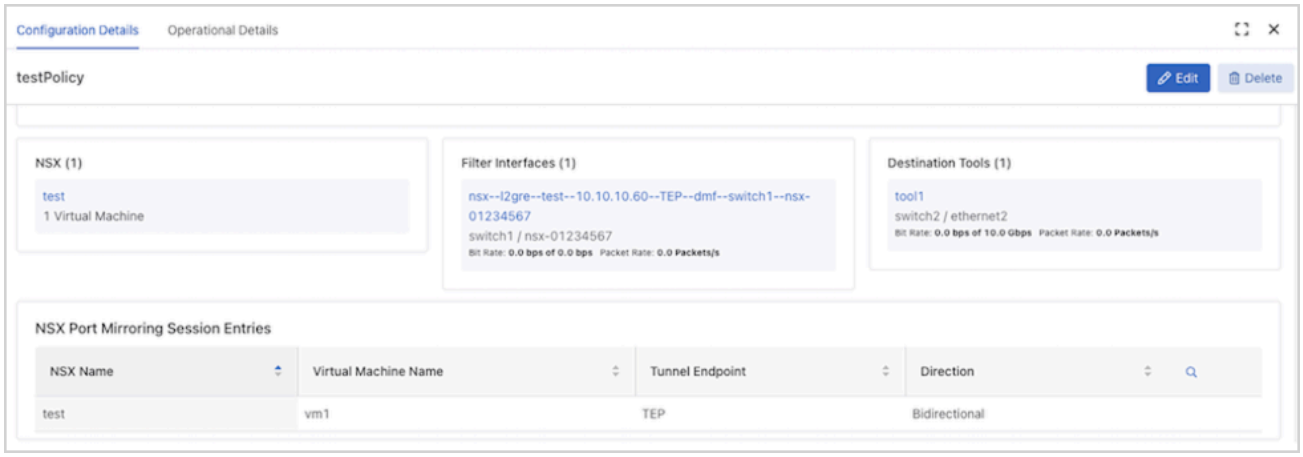
1. Navigate to **Monitoring** > **Policies**.
2. Select **Create Policy** and add the NSX instance as a traffic source.
3. Use **Add** to configure monitoring, such as the VM names to monitor, the destination tunnel endpoint, and the mirrored traffic direction. If the tunnel endpoint is omitted for a VM entry, the NSX instance's default tunnel endpoint is used as the destination.

Figure 10-7: Create Policy



Policy configuration details related to the NSX integration appear in the policy's **Configuration Details** page.

Figure 10-8: Policy Configuration Details



To edit the NSX monitoring configuration in a policy, select **1 Virtual Machine** on the **Edit Policy** page.

10.5 Show Commands

After configuring an NSX integration instance, the **show nsx *instance_name*** command displays the configuration and connection status information, including the connection status of all configured vCenter compute managers.

```
dmf-controller# show nsx test
# NSX  Hostname        State     Last Update Time               Detail State
            Version
-|----|---------------|---------|------------------------------|--------------------------------
--------------|------------------|
1 test test.arista.com connected 2025-03-28 22:36:12.975000 UTC connected; connected to compute
 managers [vcm] 4.1.2.1.0.22667789
```

The **show nsx *instance_name* detail** command displays detailed status information about the integration.

```
dmf-controller# show nsx test detail
~~~~~~~~~~~~~~~~~~~~~~~~~~ NSX ~~~~~~~~~~~~~~~~~~~~~~~~~~
NSX                      : test
Hostname                 : test.arista.com
State                    : connected
Last Update Time         : 2025-03-28 22:36:12.975000 UTC
Detail State             : connected; connected to compute managers [vcm]
Detailed Error Info      :
Version                  : 4.1.2.1.0.22667789
Hashed Password          : abc123
Username                 : admin
Preserve Mirror Sessions : False
```

The **show nsx *instance_name* alert** and **show nsx *instance_name* error** commands display runtime warnings and alerts, and errors, if any.

```
dmf-controller# show nsx instance_name alert
dmf-controller# show nsx instance_name error
```



Note: It is possible to specify **all** in each **show nsx** command to see the information for all NSX integration instances on the DMF Controller; for example, **show nsx all alert**.

The **show nsx *instance_name* compute-manager** command displays the vCenter compute managers that have been added to the NSX fabric. This information can be useful when configuring the necessary compute managers under the NSX instance configuration.

```
dmf-controller# show nsx test compute-manager
~~~~~~~~~~~~ vCenter Compute Managers  ~~~~~~~~~~~~
# NSX  Display Name Hostname/IP   vCenter Version
-|----|------------|-------------|---------------|
1 test vcm          10.10.10.10   8.0.2
```

The **show nsx *instance_name* host** command displays the host transport nodes in NSX, as well as the mirror stack VMkernel adapters configured on the transport nodes. The **show nsx *instance_name* host detail** command includes the mirror stack IP routing table entries.

```
dmf-controller# show nsx test host
~~~~~~~~~~~~~~ Host Transport Nodes ~~~~~~~~~~~~~~
# NSX vCenter Compute Manager Host Transport Node
-|----|-----------------------|-------------------|
1 test 10.10.10.10 10.10.10.20
2 test 10.10.10.10 10.10.10.30
~~~~~~~~~~~~ Mirror Stack VMkernel Adapters ~~~~~~~~~~~~
# NSX Host Transport Node Vmknic Name Vmknic IP Address
-|----|-------------------|-----------|-----------------|
1 nsx1 10.10.10.20 vmk1 10.10.10.21
2 nsx1 10.10.10.30 vmk1 10.10.10.31
3 nsx1 10.10.10.30 vmk2 10.10.10.32
~~~~~~~~~~~~ Mirror Stack IP Route Entries ~~~~~~~~~~~~
# NSX Host Transport Node IP Prefix Vmknic Name
-|----|-------------------|---------------|-----------|
1 nsx1 10.10.10.20 192.168.199.0/24 vmk1
2 nsx1 10.10.10.20 192.168.200.0/24 vmk1
3 nsx1 10.10.10.30 0.0.0.0/0 vmk1
4 nsx1 10.10.10.30 192.168.199.0/24 vmk2
5 nsx1 10.10.10.30 192.168.200.0/24 vmk2
```

The **show nsx *instance_name* endpoint** command displays the virtual machines in NSX, including their network adapters, IP addresses, and power state. The output displays the current host transport node where the VM resides, and any active mirror session that includes the endpoint.



Note: IP address information is unavailable when the VM is powered off.

```
dmf-controller# show nsx test endpoint
#  NSX  Endpoint Machine Name  Virtual Interface MAC Address                IP Address   Power
 State Mirror Sessions
--|----|--------|-------------|-----------------|--------------------------|------------|------
-----|-------------------------------------------------------------------|
1  test vm1      10.10.10.20   Network adapter 1 00:50:56:00:00:00 (VMware) 10.10.10.50  powered-
on  dmf:l2gre:bidirectional:TEP:0123456789abcdef0123456789abcdef01234567
2  test vm1      10.10.10.20   Network adapter 2 00:50:56:00:00:01 (VMware) 10.10.10.51  powered-
on  dmf:l2gre:bidirectional:TEP:0123456789abcdef0123456789abcdef01234567
3  test vm2      10.10.10.20   Network adapter 1 00:50:56:00:00:02 (VMware) 10.10.10.52  powered-
on
3  test vm3      10.10.10.30   Network adapter 1 00:50:56:00:00:03 (VMware)              powered-
off
```

The **show nsx *instance_name* session** command displays the active mirror sessions in NSX:

```
dmf-controller# show nsx test session
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ L2-GRE Mirror Sessions ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# NSX  Src VM Src Direction Tunnel Endpoint Tunnel Dst  Overall Mirror Stack Status
-|----|------|-------------|---------------|-----------|--------------------------|
1 test vm1    bidirectional TEP             10.10.10.40 success
```

The **show nsx *instance_name* session detail** command displays more detailed information about these NSX mirror sessions:

```
dmf-controller# show nsx test session detail
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ L2-GRE Mirror Sessions
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# NSX  Session Name                                                    Src VM Src Direction
 Tunnel Endpoint Tunnel Dst  GRE Key  Overall Mirror Stack Status
```

```
-|----|-------------------------------------------------------------------|------|-------------|
---------------|-----------|--------|---------------------------|
1 test dmf:l2gre:bidirectional:TEP:0123456789abcdef0123456789abcdef01234567 vm1    bidirectional
 TEP             10.10.10.40 32118876 success

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Detailed Mirror Stack Status  ~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# NSX  Session Name                                                          Transport Node Mirror
 Stack Status Mirror Stack VMkernel Adapter Status Status Detail
-|----|-------------------------------------------------------------------|-------------|------
-------------|-----------------------------------|---------------------------------------|
1 test dmf:l2gre:bidirectional:TEP:0123456789abcdef0123456789abcdef01234567 10.10.10.20    success
             success                                Both mirror stack and vmknic are HEALTHY
```

NSX configuration-related errors can be displayed using the **`show fabric errors nsx-error`** command:

```
dmf-controller# show fabric errors nsx-error
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ NSX related error  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# NSX Name Error
-|--------|-----------------------------------------------------------------------------------------|
1 test     NSX instance test config references a default tunnel endpoint TEP that does not exist.
```

NSX policy-related errors can be displayed using the **`show fabric errors policy-error`** command:

```
dmf-controller# show fabric errors policy-error
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Policy related error ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Error                                                                  Policy Name
-|----------------------------------------------------------------------|-----------|
1 NSX instance test entry references a VM vm4 that does not exist.      testPolicy
```

NSX policy-related warnings can be displayed using the **`show fabric warnings policy-warning`** command:

```
dmf-controller# show fabric warnings policy-warning
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Policy related warning ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Policy Name Warning
-|-----------|-----------------------------------------------------------------------|
1 testPolicy  NSX instance test entry references a VM vm3 that is not powered on.
```

The relevant NSX errors or warnings for a given NSX policy can also be found in the Vendor Integration Alerts table of the **`show policy`** ***`policy_name`*** command:

```
dmf-controller# show policy testPolicy
Policy Name                          : testPolicy
Config Status                        : active - forward
Runtime Status                       : installed
...
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Vendor Integration Alert(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Priority Vendor Message
-|--------|------|-------------------------------------------------------------------------|
1 error    NSX    NSX instance test entry references a VM vm4 that does not exist.
2 warning  NSX    NSX instance test entry references a VM vm3 that is not powered on.
```

10.6 Troubleshooting

- If the expected mirror stack VMkernel adapter information is not being displayed for a host transport node, despite this VMkernel adapter being configured correctly on vCenter, ensure that the vCenter compute manager for the given host transport node is configured under the NSX instance configuration and that DMF has connected successfully to the vCenter.
- If any fabric errors specify a missing mirror session, user action on NSX may be required. From the NSX Manager GUI, navigate to **Plan & Troubleshooting** > **Port Mirroring**. If the expected mirror session config is present, select the **Source** entry for the session. Selecting the **Failed** entry under the **Status** column will indicate why the mirror session creation failed.



Note: Mirror session creation may initially fail on NSX, but the NSX Manager may recover and successfully create the mirror session later. This fabric error is cleared and the mirror session info is then visible from DMF. If the mirror session config is absent on NSX, delete the NSX monitoring config from the DMF policy and reconfigure it, or delete the policy and recreate it.

10.7 Considerations

- Modifying or deleting auto-generated filter interfaces for NSX integration or adding them manually to non-NSX policies will result in undefined behavior.
- Modifying or deleting DMF-managed mirroring profiles, source groups, or destination groups on the NSX manager will result in undefined behavior.
- Currently, a DMF policy can only use one of the NSX or CloudVision integrations as a traffic source.
- Only VM network adapters attached to NSX logical ports can be monitored using the NSX integration. When monitoring a VM with multiple network adapters, some of which are not attached to NSX logical ports, traffic is mirrored only on the network adapters attached to NSX logical ports. vCenter VMs with no network adapters attached to NSX logical ports will not be displayed and must be monitored using the vCenter integration instead.
- Only port mirroring using a mirror stack VMkernel adapter is supported to reduce the impact of mirror sessions on production traffic in the NSX fabric.
- Mirror sessions may not be retained across Controller restart/failover or VM migration (VMware vMotion).

10.8 Resources

Refer to Managing DMF Policies and Tunneling Between Data Centers for further information on policies and tunneling.

Chapter 11

CloudVision DMF Integration

This chapter describes integrating CloudVision with the DANZ Monitoring Fabric (DMF).

Overview

Compatibility Requirements

CloudVision DMF Integration

Troubleshooting

Considerations

Resources

Using the Command Line Interface

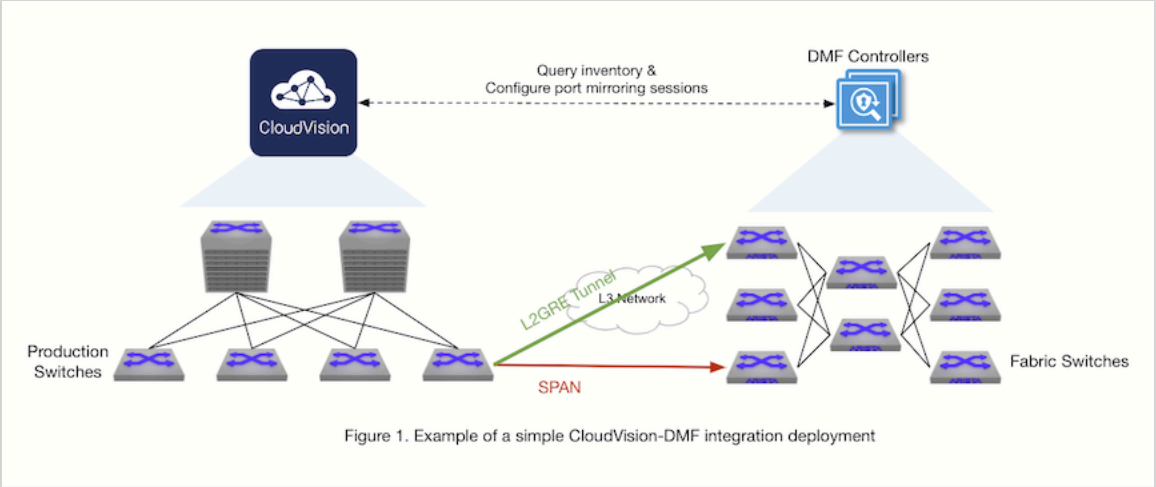
11.1 Overview

In a typical CloudVision-DMF integration deployment, CloudVision Portal (CVP) deploys alongside the DANZ Monitoring Fabric (DMF). The DMF Controller communicates with CVP to retrieve its managed device inventory and configures port mirroring sessions on any CVP-managed production devices that are Arista Extensible Operating System (EOS) switches.

Configuration on the DMF Controller provides the information necessary to communicate with CVP: the CVP hostname or IP address and user credentials.

Policy configuration on the DMF Controller specifies what to monitor in the production network managed by CVP, such as the production switches, the switch interfaces to monitor traffic from, and the direction of mirrored traffic (bidirectional, ingress, or egress). In addition, the configuration on the DMF Controller can define whether to use a Switch Port Analyzer (SPAN) session or a Layer-2 Generic Routing Encapsulation (L2GRE) tunnel session on a CVP-managed device. When using SPAN, the DMF configuration includes the switch interface to monitor traffic. When using L2GRE, the DMF configuration includes monitoring traffic to the Tunnel End Point (TEP).

Figure 11-1: Simple CloudVision-DMF Integration Deployment



The preceding figure illustrates a simple CloudVision-DMF integration configuration where CloudVision Portal and DMF can communicate with each other. DMF monitors traffic from CVP directly to one of its fabric switches (a filter switch), as indicated by the red arrow labeled “SPAN.” DMF also monitors traffic from CVP to a TEP configured on the fabric using an L2GRE tunnel, as indicated by the green arrow labeled “L2GRE Tunnel". Since DMF initiates monitoring using policy configuration, the policies monitoring CVP will handle the traffic according to their configuration, for example, forwarding it to a delivery interface. This feature enables the automation of the creation, modification, and deletion of filter interfaces and tunnel interfaces in DMF and mirroring sessions on CVP-managed devices.

11.2 Compatibility Requirements

EOS Platform Compatibility

- Any platform supporting port mirroring or mirroring to a GRE tunnel specifying the GRE key is supported; refer to EOS Port Mirroring TOI and EOS Mirroring to GRE Tunnel TOI for more details.

CloudVision Compatibility

- On-premise 2024.2.0 and newer is recommended.

CloudVision Requirements

- The user configured in DMF for CloudVision Portal (CVP) integration must have sufficient permissions in CVP. The minimum permissions required are:
  - Devices: Read access to inventory management.
  - gNMI: Read and write access to the gNMI service.
  - The **`cvpadmin`** username must be configured on the switches for the change control to go through while configuring a policy or monitoring session. This is done on the production switches managed by the CVP.

    ```
    username cvpadmin privilege 15 secret arastra
    aaa authorization exec default local
    ```

- Register the devices that DMF will monitor for use in Studios using the Inventory and Topology Studio.
- For L2GRE mirroring, the switch must have IP routing enabled. For some switch platforms, the GRE payload will not include the terminated headers of mirrored packets by default, requiring additional switch configuration to obtain the full payload. Refer to the **GRE payload type** section of the EOS Mirroring to GRE Tunnel TOI for further details.

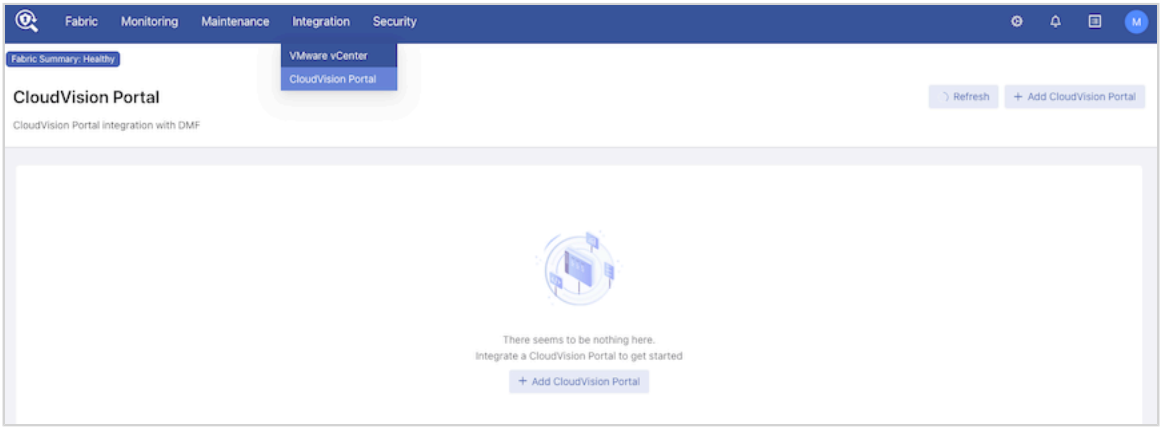
11.3 CloudVision DMF Integration

To integrate with CloudVision Portal, configure a CVP instance in the DMF Controller, enabling communication between CVP and DMF. The CVP hostname or IP address must be reachable from the DMF Controller. If CVP is a multi-node system, using a hostname that will resolve to the primary node is recommended to maintain the connection in case of a primary node failure. The user in the CVP integration configuration must have at least the permissions in CloudVision as outlined in CloudVision Requirements.

Navigate to **Integration** > **CloudVision Portal**.

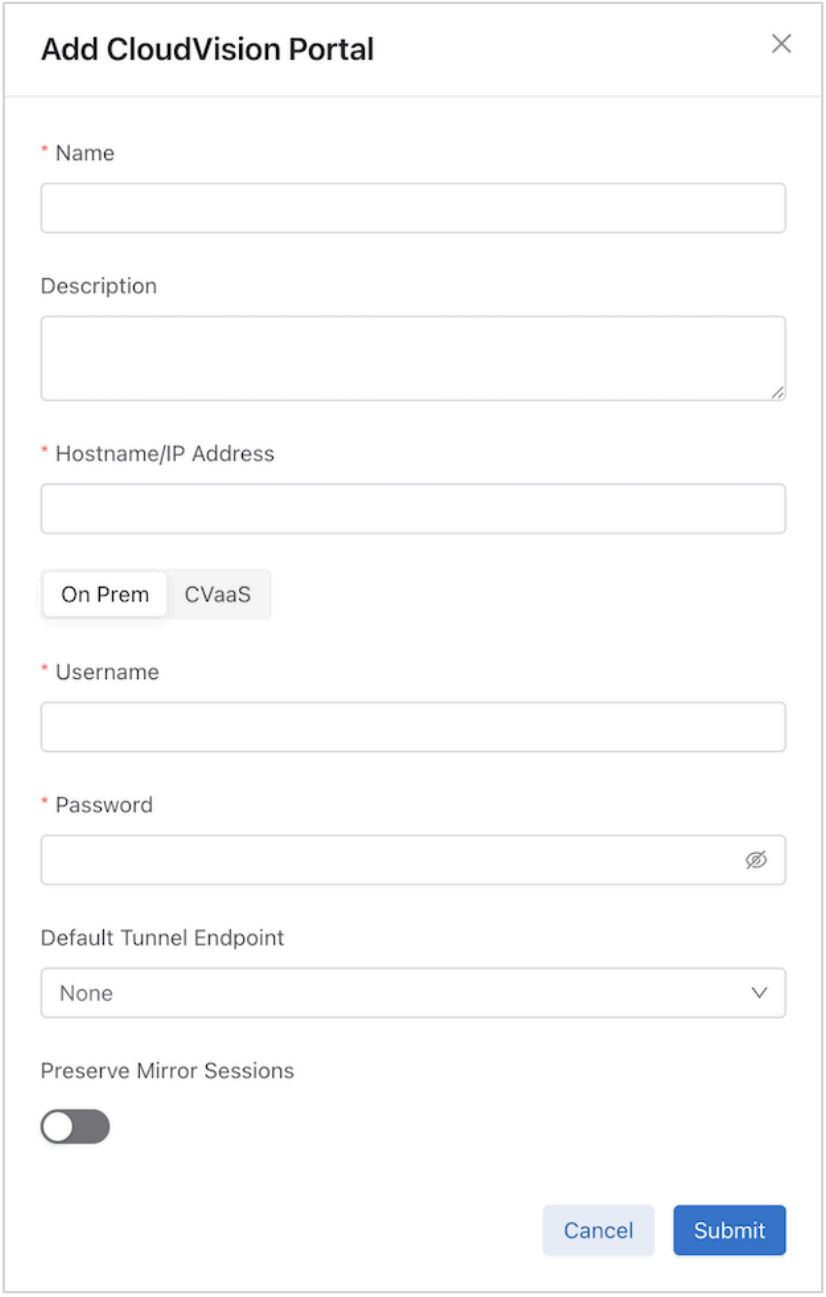
Select **Add CloudVision Portal** to create a new CVP integration instance.

Figure 11-2: DMF Integration CloudVision Portal



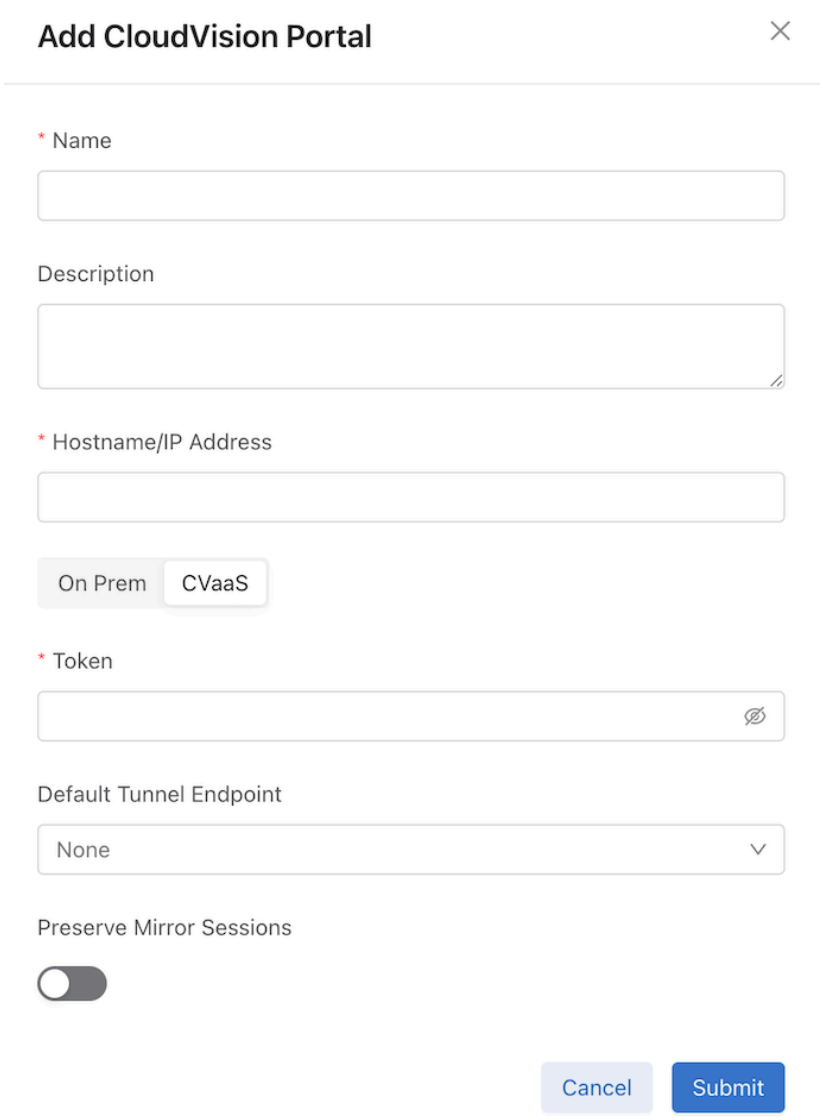
Enter the **On Prem** CVP integration instance configuration details and select **Submit**.

Figure 11-3: Add On Prem CloudVision Portal



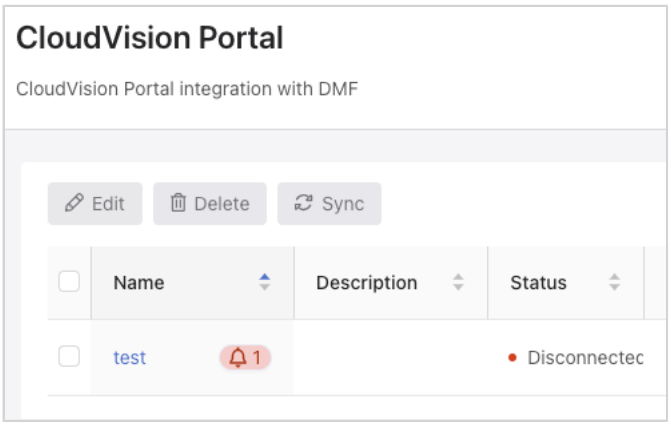
Select **CVaaS** to add a CVaaS integration instance and enter the configuration details and select **Submit**.

Figure 11-4: Add CVaaS CloudVision Portal



If there are any warnings or errors with the CVP integration instance, use the **alarm bell** icon to view more details.

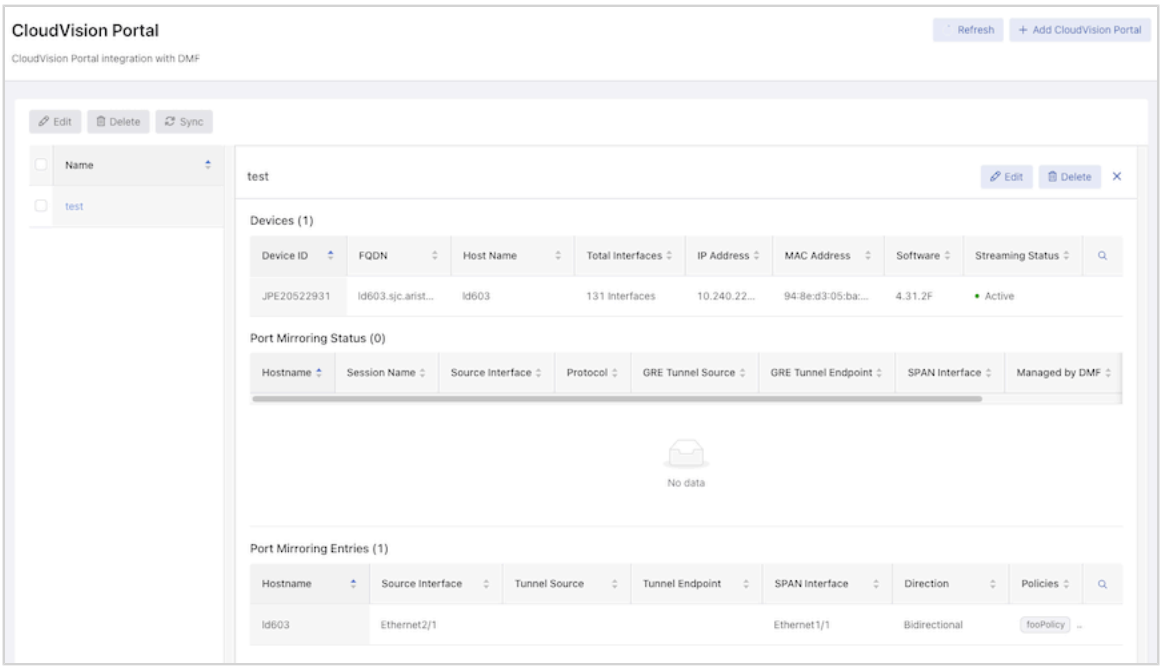
Figure 11-5: CloudVision Portal



Select the **CVP instance name** to view its details, including the device inventory, port mirroring status, and port mirroring entries in DMF policies.

Note: The **Port Mirroring Entries** table will not be visible if no policies using this CVP instance as a traffic source have been configured.

Figure 11-6: CloudVision Portal Dashboard

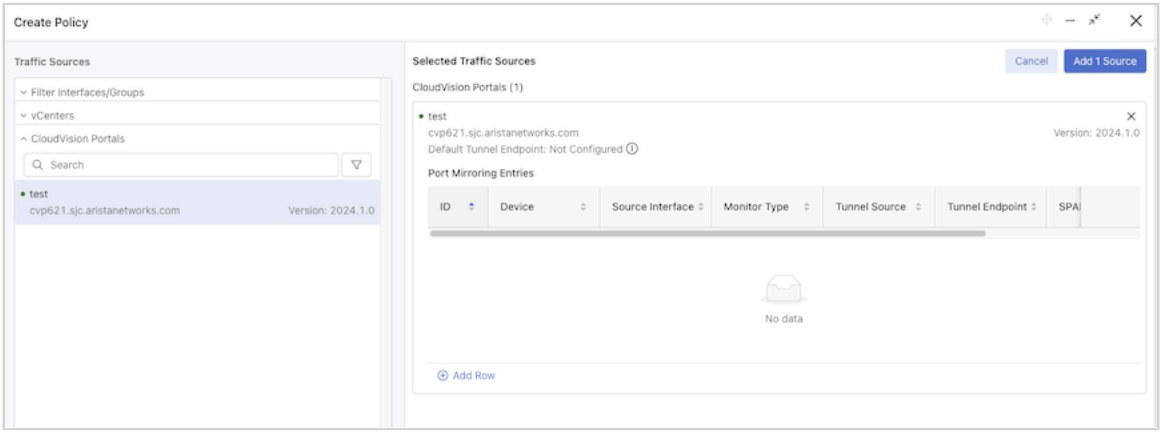


To start monitoring traffic from the production network:

1. Navigate to **Monitoring** > **Policies**.
2. Select **Create Policy** and add CVP instances as traffic sources.

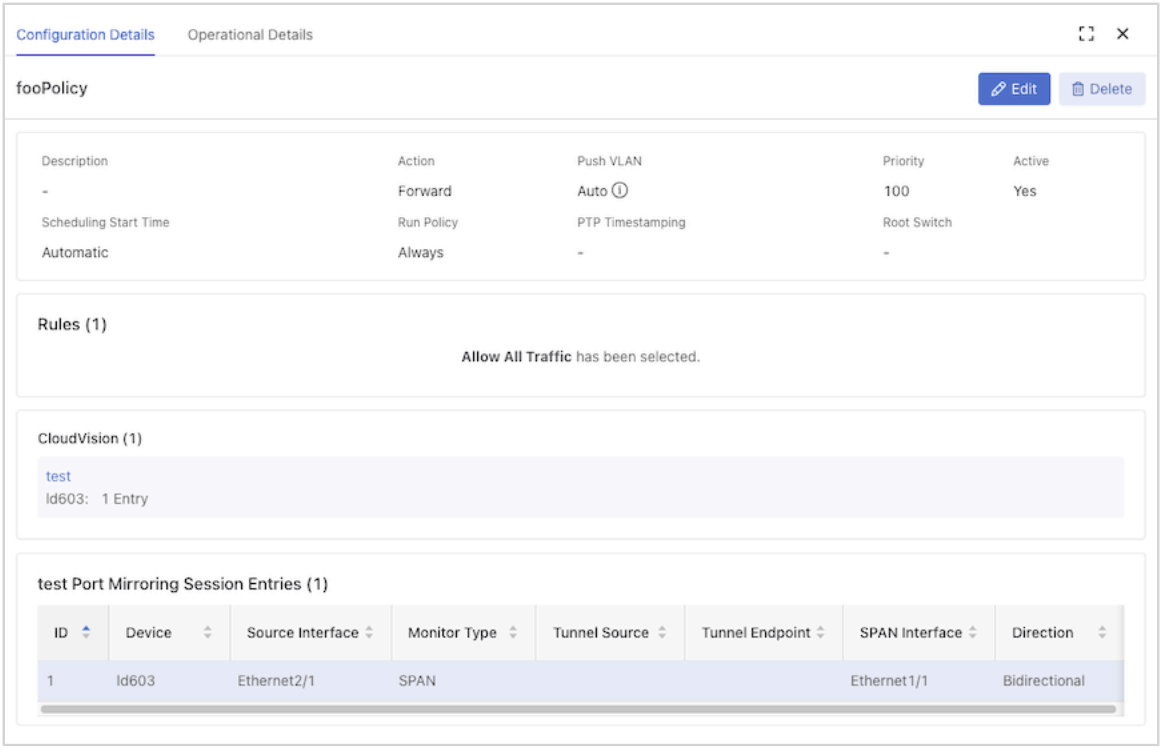
3. Select **Add Row** to configure monitoring, such as the device interfaces to monitor, the monitor type (e.g., SPAN or L2GRE), the mirrored traffic direction, and the destination.

Figure 11-7: Create DMF Policy



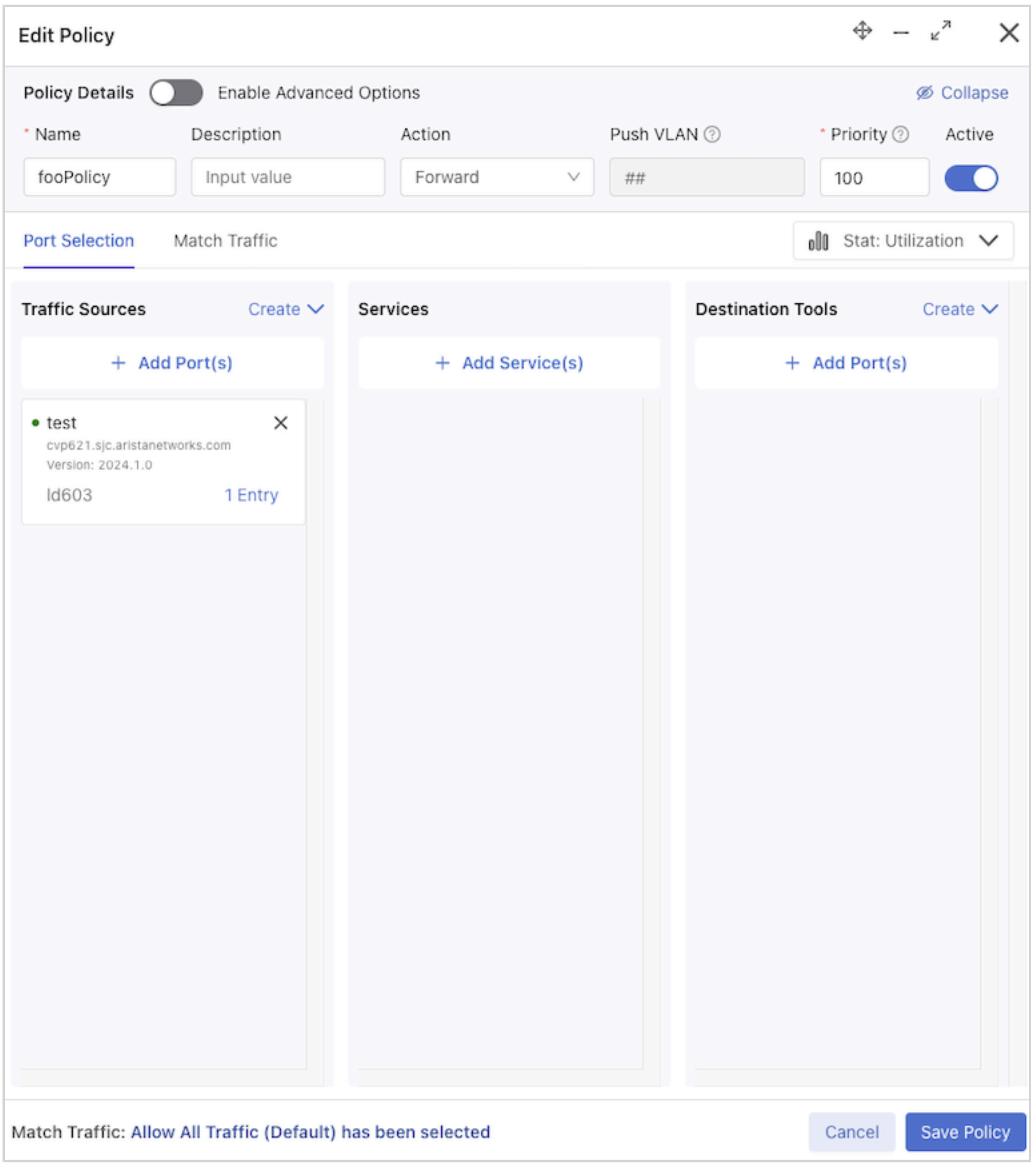
Policy configuration details related to CVP integration appear in the policy's **Configuration Details** page.

Figure 11-8: Configuration Details



To edit the CVP monitoring configuration in a policy, use **Edit** and select **1 Entry** on the **Edit Policy** page, make the required changes, and select **Save Policy**.

Figure 11-9: Edit Policy



11.4 Troubleshooting

- If there are any fabric errors in creating a mirroring session on a CVP-managed device that specify that user action in CVP is required, take the appropriate corrective action in CVP. Next, delete the CVP monitoring config from the policy and reconfigure it, or delete the policy and then recreate it.
- When using username/password authentication, if there is a message in the **`show cvp alert`** command output stating a session failed to be updated and an authentication error message in `/var/log/vsphere-extension/vsphere-extension.log` containing `GNMI7001…Status unauthenticated`, run the **`sync cvp`** command, and then deactivate and reactivate the relevant DMF policy.
- When using service account token authentication, if there is a token-related error in the **`show cvp`** command output or a message in the **`show cvp alert`** command output stating a session failed to be updated, ensure that the token has not been deleted from the service account, and that the service account has not had gNMI service read/write permissions revoked. In the former case, configure a new valid token for the CVP instance; in the latter case, update the permissions for the service account and run the **`sync cvp`** command. Then, delete the DMF policy and recreate it.
- When using service account token authentication, if there is a fabric error for a missing mirroring session, check that the CVP instance is connected and that the token has not expired with the **`show cvp detail`** command. If the token has expired, configure a new valid token for the CVP instance, then delete the DMF policy and recreate it.
- If a missing mirroring session fabric error occurs, a possible cause is that the device had not been registered for use in Studios in CVP using the Inventory and Topology studio. If this is the case, register the device in CVP, and then delete the DMF policy and recreate it.

11.5 Considerations

The following limitations apply to the DANZ Monitoring Fabric (DMF) Controller and CloudVision integration.

- Modifying or deleting auto-generated filter interfaces for CVP integration or adding them manually to non-CloudVision policies will result in unexpected behavior.
- If two or more DMF policies have overlapping CVP monitoring configurations with the same destination, the configuration should be the same in these policies if unexpected traffic is undesired. For example, if a policy has source interfaces A and B on a device in a CVP instance with a SPAN interface as the destination, and if another policy has one source interface, A, on the same device in the same CVP instance with the same SPAN interface as the former policy, then the latter policy will unexpectedly receive traffic from B as well as the expected source A. This is because the mirroring session in the production switch is reused for both policies.
- An error state in the CVP deployment may prevent DMF from configuring mirroring sessions on the production switches. If DMF encounters such a failure, a fabric error is displayed, and user action in CVP is required. Examples of such a state include the production switch not being in compliance in CVP, in which case it needs to be brought into compliance, or if there are any pending change controls in CVP, they must be addressed. After taking the appropriate corrective action in CVP, deactivate and reactivate the DMF policy with the error (or delete and recreate it) to retry configuring mirroring.

11.6 Resources

Please refer to tunneling and policies in this guide for further information.

11.7 Using the Command Line Interface

CloudVision DMF Integration

11.7.1 CloudVision DMF Integration

To integrate with CloudVision Portal, configure a CVP instance in the DMF Controller, enabling communication between CVP and DMF. The CVP hostname or IP address must be reachable from the DMF Controller. If CVP is a multi-node system, using a hostname that will resolve to the primary node is recommended to maintain the connection in case of a primary node failure. The user in the CVP integration configuration must have at least the permissions in CloudVision as outlined in CloudVision Requirements.

Configure using the CLI

Configure a CVP instance on the DMF Controller using the following series of commands:

```
dmf-controller(config)# cvp cvp_instance_name
dmf-controller(config-cvp)# host-name cvp_hostname_or_ip
dmf-controller(config-cvp)# username username
dmf-controller(config-cvp)# password password
```

Starting with the DMF 8.8.0 release, a token field is introduced for DMF-CV integration, allowing authentication support using service account tokens. For information on obtaining authentication tokens on CV, refer to the Help Center or the Service Accounts TOI.

```
dmf-controller(config-cvp)# token service-account-token
```



Note: The username and password authentication is only supported for configuring **On Prem** CV instances and is not available for **CVaaS** instances.

Add a description to the CVP instance using the **description** command, as required.

```
dmf-controller(config-cvp)# description description_of_cvp_instance
```

Refresh the connection between DMF and CVP with the **sync** command, which sends a request to CVP to re-authenticate the connection and to re-fetch the inventory:

```
dmf-controller(config)# sync cvp cvp_instance_name
```

To use L2GRE tunnels in the integration, enable tunneling in the DMF Controller. For SNL switches, set the match mode to one of the following that is compatible with tunneling: **full-match** or **l3-l4-offset-match**. For EOS switches, tunneling will work in any match mode. Configure tunnel endpoints to allow monitoring from CVP to DMF using an L2GRE tunnel; add a tunnel endpoint to a policy configuration or to a CVP integration instance's configuration to ***optionally*** define a default tunnel endpoint for this instance.

To configure a tunnel endpoint for EOS switches, use the following command:

```
dmf-controller(config)# tunnel-endpoint tep_name switch fabric_switch fabric_switch_interface ip-address tep_ip mask subnet_mask gateway gateway_ip
```

For SWL switches, the **mask** and **gateway** parameters are optional in the tunnel-endpoint command:

```
dmf-controller(config)# tunnel-endpoint tep_name switch fabric_switch fabric_switch_interface ip-address tep_ip
```

To set a default tunnel endpoint for a CVP integration instance, use the following commands:

```
dmf-controller(config)# cvp cvp_instance_name
dmf-controller(config-cvp)# default-tunnel-endpoint tep_name
```

To remove a default tunnel endpoint for a CVP integration instance, use the following commands:

```
dmf-controller(config)# cvp cvp_instance_name
dmf-controller(config-cvp)# no default-tunnel-endpoint tep_name
```

Starting with the DMF 8.6.0 release, a configuration flag called **preserve-mirror-sessions** per CVP instance indicates whether mirroring sessions are preserved for the CVP instance when uninstalling DMF policies configured with it. By default, the flag is **false**, meaning existing mirroring sessions are automatically removed if the relevant DMF policies are uninstalled.

Enable preserving mirroring sessions using the **preserve-mirror-sessions** command.

```
dmf-controller(config-cvp)# preserve-mirror-sessions
```

Conversely, disable preserving mirroring sessions (default behavior) using the **no preserve-mirror-sessions** command.

```
dmf-controller(config-cvp)# no preserve-mirror-sessions
```

Monitoring Configuration in Policies

DMF uses policies to create, update, or remove the monitoring of CVP-managed devices. DMF supports monitoring multiple CVP instances, switches, and interfaces as mirroring sources in a single policy or across policies. Configure the mirrored traffic direction to one of the following settings:

- bidirectional (default)
- ingress
- egress

After enabling CVP integration in a DMF policy (i.e., adding a CVP instance as a traffic source), the DMF Controller will automatically create filter interfaces and tunnel interfaces, with origination "auto-generated." A mirroring session is automatically created on the CVP-managed switch; DMF does this via the mirroring Studio and the change control process on CVP.



Note: If a DMF-managed mirroring session exists on a switch for one DMF policy with identical sources and the same destination as needed for another DMF policy, both policies use the same mirroring session.

Add a CVP instance as a traffic source in a DMF policy using the following series of commands:

```
dmf-controller(config)# policy policy_name
dmf-controller(config-policy)# filter-cvp cvp_instance_name
dmf-controller(config-policy-cvp)#
```

To monitor traffic using SPAN, configure a SPAN interface (on the CVP-managed device) as the destination in a DMF policy, along with the source interfaces (on the CVP-managed device) and optionally the direction for each source interface.

Note: LLDP must be enabled on the SPAN interface before it can be used in a SPAN mirror session.

Configure the switch interfaces on CVP-managed devices individually as source interfaces to a SPAN interface on that switch using the following series of commands where including the direction is optional:

```
dmf-controller(config-policy-cvp)# device device_hostname
dmf-controller(config-policy-cvp-device)# src-interface source_interface
span-interface span_interface direction ingress | egress | bidirectional
```

Configure the switch interfaces on CVP-managed devices as source interfaces using an interface range to a SPAN interface on that switch using the following series of commands where including the direction is optional:

```
dmf-controller(config-policy-cvp)# device device_hostname
dmf-controller(config-policy-cvp-device)# src-interface-range start start_of_range
end end_of_range span-interface span_interface direction ingress | egress | bidirectional
```

To monitor traffic using L2GRE tunneling, choose from two options: (1) configure a tunnel endpoint (in DMF) as the destination in a DMF policy along with the source interfaces (on the CVP-managed device) and optionally the direction for each source interface, or (2) omit the destination in a DMF policy along with configuring the source interfaces (on the CVP-managed device) and optionally the direction for each source interface.

A GRE tunnel source IP can be optionally configured on DMF as the tunnel source IP on the CVP-managed device to overcome reachability issues due to possible Reverse Path Forwarding (RPF) checks between the CVP and DMF deployment. By default, the switch's management IP is used as the tunnel source IP.

Configure the switch interfaces on CVP-managed devices individually as source interfaces to a tunnel endpoint configured in DMF using the following series of commands where the direction is optional:

```
dmf-controller(config-policy-cvp)# device device_hostname
dmf-controller(config-policy-cvp-device)# src-interface source_interface gre-tunnel-src src_ip
gre-tunnel-endpoint tep_name direction ingress | egress | bidirectional
```

The **src-interface-range** command is also supported for GRE tunnel configuration in a policy.

The following example illustrates two DMF policies' configuration, where **testPolicy1** is monitoring traffic from **Ethernet1** on the CVP-managed device (production switch) called **dev1** in the CVP instance, **test**, to **Ethernet2** on the same device, using SPAN, and forwarding the traffic to the delivery interface called **tool1**; **testPolicy2** is monitoring traffic from **Ethernet5** on the CVP-managed device called **dev2** in the same CVP instance, **test**, to the default tunnel endpoint called **TEP1** defined in the CVP integration instance configuration, using L2GRE tunneling, and forwarding the traffic to the delivery interface called **tool2**.

```
! cvp
cvp test
  default-tunnel-endpoint TEP1
  hashed-password abc123
  host-name test.arista.com
  user-name cvpadmin

! policy
policy testPolicy1
   action forward
   delivery-interface tool1
   1 match any
   filter-cvp test
```

```
     !
     device dev1
     src-interface Ethernet1 span-interface Ethernet2

policy testPolicy2
  action forward
  delivery-interface tool2
  1 match any
  filter-cvp test
    !
    device dev2
      src-interface Ethernet5
```

Suppose you remove the configuration to monitor CVP-managed devices from the DMF Controller. In that case, the system removes the corresponding auto-generated filter interfaces and tunnel interfaces from DMF and deletes the auto-created mirroring sessions on the switch.

To stop monitoring a source interface or a range of source interfaces on a CVP-managed device, remove its configuration from a DMF policy using the following series of commands:

```
dmf-controller(config-policy-cvp-device)# no src-interface source_interface
dmf-controller(config-policy-cvp-device)# no src-interface-range start start_of_range
 end end_of_range
```

Stop monitoring a device in a CVP instance in a DMF policy.

```
dmf-controller(config-policy-cvp)# no device device_hostname
```

Stop monitoring a CVP instance in a DMF policy and all its devices.

```
dmf-controller(config)# policy policy_name
dmf-controller(config-policy)# no filter-cvp cvp_instance_name
```

To disable integration with a CVP instance, remove its CVP integration instance configuration and remove it from all DMF policies using the following series of commands:

```
dmf-controller(config)# no cvp cvp_instance_name
dmf-controller(config)# policy policy_name
dmf-controller(config-policy)# no filter-cvp cvp_instance_name
```

Show Commands

After configuring a CVP instance, the **show cvp *cvp_instance_name*** command displays the configuration and connection status information.

```
dmf-controller(config)# show cvp test
# CVP  Hostname        State     Last Update Time               Detail State Version
-|----|---------------|---------|------------------------------|------------|--------|
1 test test.arista.com connected 2023-12-13 05:17:28.512000 UTC connected    2024.1.0
```

The **show cvp *cvp_instance_name* detail** command displays detailed status information about the integration.

```
dmf-controller(config)# show cvp test detail
CVP                     : test
Hostname                : test.arista.com
State                   : connected
Last Update Time        : 2023-12-13 05:17:54.072000 UTC
Detail State            : connected
Version                 : 2024.1.0
Preserve Mirror Sessions : False
```

The earlier command includes token expiry time when using service account token based authentication. It also displays the **Organization** (tenant) **Name** when using CVaaS.

```
dmf-controller(config)# show cvp test detail
CVP                       : test
Hostname                  : www.arista.io
State                     : connected
Last Update Time          : 2025-05-30 04:22:39.382000 UTC
Detail State              : connected
Detailed Error Info       :
Version                   : CVaaS
Preserve Mirror Sessions : False
Organization Name         : test-org
Token Expiry Time         : 2025-06-01 06:59:59 UTC
```

The **show cvp *cvp_instance_name* alert** and **show cvp *cvp_instance_name* error** commands display runtime warnings and alerts, and errors, if any.

```
dmf-controller(config)# show cvp cvp_instance_name alert
dmf-controller(config)# show cvp cvp_instance_name error
```



Note: It is possible to specify **all** in the above **show cvp** commands to see the information for all CVP integration instances on the DMF Controller; for example, **show cvp all alert**.

The **show cvp *cvp_instance_name* device *device_hostname*** command displays the device inventory in the CVP deployment; only EOS devices are supported. Using **all** is possible for the CVP instance name and the device hostname in the **show cvp *cvp_instance_name* device *device_hostname*** command.

```
dmf-controller(config)# show cvp test device all
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Device Inventory ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# CVP  Device FQDN              Streaming Model   Software IP Address   MAC Address                Device ID
-|----|------|-----------------|---------|-------|--------|------------|--------------------------|-----------|
1 test dev123 dev123.arista.com active    ABC-123 4.31.2F  10.10.10.10  aa:bb:cc:dd:ee:ff (Arista) DEV123
```

The **show cvp *cvp_instance_name* device *device_hostname* interface** command includes a list of all the device interfaces.

```
dmf-controller(config)# show cvp test device all
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Device Inventory ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# CVP  Device FQDN              Streaming Model   Software IP Address   MAC Address                Device ID
-|----|------|-----------------|---------|-------|--------|------------|--------------------------|-----------|
1 test dev123 dev123.arista.com active    ABC-123 4.31.2F  10.10.10.10  aa:bb:cc:dd:ee:ff (Arista) DEV123

~~~~ Device Interfaces ~~~~
#  CVP  Device Interface
--|----|------|-----------|
1  test dev123 Ethernet1
2  test dev123 Ethernet2
3  test dev123 Ethernet3
```

After configuring a DMF policy to take a CVP instance as a traffic source, the **show fabric errors** command displays any errors with the integration relating to monitoring, if any. This command existed before DMF 8.5.0 and has been enhanced for this feature.

```
dmf-controller(config)# show fabric errors
```

In addition, the DMF Controller can show the current mirroring sessions configured on a CVP-managed device used to confirm the current state of a mirroring session created by DMF (thus, managed by DMF) or otherwise (non-DMF-managed sessions are only displayed in the **detail** command). There are three commands to display the mirroring state on a CVP-managed device in varying levels of detail, as follows:

1) The **show cvp *cvp_instance_name* device *device_hostname* session** command displays only the mirroring sessions managed by DMF.

```
dmf-controller(config)# show cvp test device dev123 session
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ L2-GRE Mirroring Sessions ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# CVP  Hostname Tunnel Endpoint Programmed in hardware Tunnel Src Tunnel Dst Src Interface Src Link Status Src Direction
-|----|--------|---------------|----------------------|----------|----------|-------------|---------------|-------------|
1 test dev123   unknown                                3.3.3.3    4.4.4.4    Ethernet2     unspecified     bidirectional

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ SPAN Mirroring Sessions  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# CVP  Hostname SPAN Interface SPAN Status Programmed Src Interface Src Link Status Src Direction
-|----|--------|--------------|-----------|----------|-------------|---------------|-------------|
1 test dev123   Ethernet5      up          true       Ethernet4     up              bidirectional
```

2) The **show cvp *cvp_instance_name* device *device_hostname* session brief** command displays a summary of the state of the mirroring sessions managed by DMF.

```
dmf-controller(config)# show cvp cvp_instance_name device device_hostname session brief
```

3) The **show cvp *cvp_instance_name* device *device_hostname* session detail** command displays all the mirroring sessions on the device, both managed by DMF and otherwise, as well as the name of each session.

```
dmf-controller(config)# show cvp cvp_instance_name device device_hostname session detail
```

Chapter 12

Advanced Fabric Settings

This chapter describes fabric-wide configuration options required in advanced use cases for deploying DMF policies.

12.1 Configuring Advanced Fabric Settings

To navigate the **DMF Features** Page, select the **gear** icon in the navigation bar.

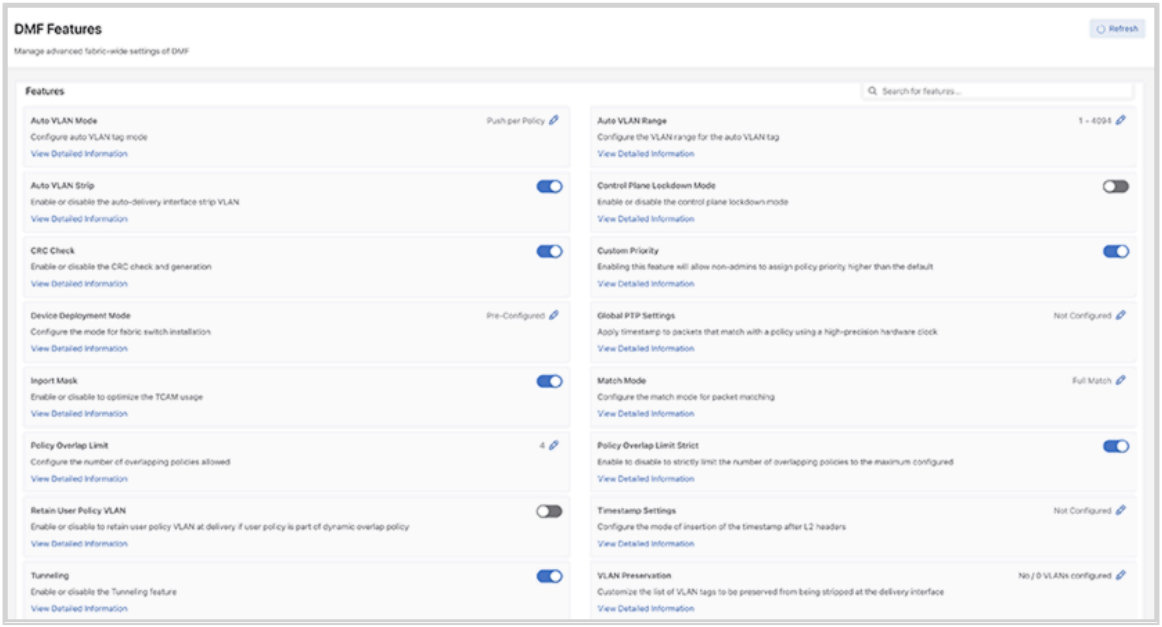
Figure 12-1: Gear Icon



Page Layout

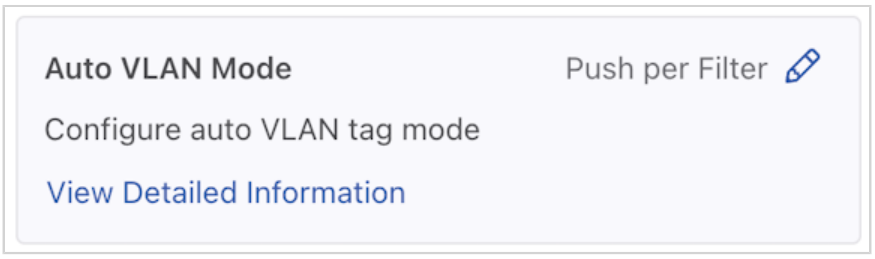
All fabric-wide configuration settings required in advanced use cases for deploying DMF policies appear in the new DMF Features Page.

Figure 12-2: DMF Features Page



Each card on the page corresponds to a feature set.

Figure 12-3: Feature Set Card Example



The UI displays the following:

- Feature Title
- A brief description
- View / Hide detailed information link
- Current Setting
- Edit Link - Use **Edit** (**pencil**) icon to change the value.

The fabric-wide options used with DMF policies include the following:

Feature Set	
Auto VLAN Mode	Auto VLAN Range
Auto VLAN Strip	CRC Check
Custom Priority	Device Deployment Mode
Inport Mask	Match Mode
Policy Overlap Limit	Policy Overlap Limit Strict
PTP Timestamping	Retain User Policy VLAN
Tunneling	VLAN Preservation

12.2 Managing VLAN Tags in the Monitoring Fabric

Analysis tools often use VLAN tags to identify the filter interface receiving traffic. How VLAN IDs are assigned to traffic depends on which auto-VLAN mode is enabled. The system automatically assigns the VLAN ID from a configurable range of VLAN IDs, from ***1*** to ***4094*** by default. Available auto-VLAN modes behave as follows:

- **push-per-policy (default)**: Automatically adds a unique VLAN ID to all traffic selected by a specific policy. This setting enables tag-based forwarding.
- **push-per-filter**: Automatically adds a unique VLAN ID from the default auto-VLAN range (***1-4094***) to each filter interface. A custom VLAN range can be specified using the **`auto-vlan-range`** command. Manually assign any VLAN ID not in the auto-VLAN range to a filter interface.

The VLAN ID assigned to policies or filter interfaces remains unchanged after controller reboot or failover. However, it changes if the policy is removed and added back again. Also, when the VLAN range is changed, existing assignments are discarded, and new assignments are made.

The push-per-filter feature preserves the original VLAN tag, but the outer VLAN tag is rewritten with the assigned VLAN ID if the packet already has two VLAN tags.

The following table summarizes how VLAN tagging occurs with the different auto-VLAN modes:

Table 7: VLAN Tagging Across VLAN Modes

Traffic with VLAN tag type	push-per-policy Mode (Applies to all supported switches)	push-per-filter Mode (Applies to all supported switches)
Untagged	Pushes a single tag	Pushes a single tag
Single tag	Pushes an outer (second) tag	Pushes an outer (second) tag
Two tags	Pushes an outer (third) tag. Except on T3-based switches, it rewrites the outer tag. Due to this outer customer VLAN is replaced by DMF policy VLAN.	Rewrites the outer tag. Due to this outer customer VLAN is replaced by DMF filter VLAN.



Note: When enabling push-per-policy, the **`auto-delivery-interface-vlan-strip`** feature is enabled (if disabled) before enabling push-per-policy. When enabling push-per-filter, the global delivery strip option is not enabled if previously disabled.

The following table summarizes how different auto-VLAN modes affect the applications and services supported.



Note: Matching on untagged packets cannot be applied to DMF policies when in push-per-policy mode.

Table 8: Auto-VLAN Mode Comparison

Auto-VLAN Mode	Supported Platform	TCAM Optimization in the Core	L2 GRE Tunnels Support	Q-in-Q Packets Preserve Both Original Tags	Support DMF Service Node Services	Manual Tag to Filter Interface
Push-per-policy (default)	All	Yes	Yes	Yes	All	Policy tag overwrites manual
Push-per-filter	All	No	Yes	No	All	Configuration not allowed

Note: Tunneling is supported with full-match or offset-match modes but not with l3-l4-match mode.

Tag-based forwarding, which improves traffic forwarding and reduces TCAM utilization on the monitoring fabric switches, is enabled only when choosing the push-per-policy option.

When the mode is push-per-filter, the VLAN that is getting pushed or rewritten can be displayed using the **`show interface-names`** command as shown below:

```
controller-1> show interface-names
~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#  DMF IF                    Switch IF     Name         Dir State Speed  VLAN Tag Analytics Ip address Connected Device
--|-------------------------|-------------|------------|---|-----|------|--------|---------|----------|----------------|
1  TAP-PORT-eth1             FILTER-SW1    ethernet1    rx  up    10Gbps 5        True
2  TAP-PORT-eth10            FILTER-SW1    ethernet10   rx  up    10Gbps 10       True
3  TAP-PORT-eth12            FILTER-SW1    ethernet12   rx  up    10Gbps 11       True
4  TAP-PORT-eth14            FILTER-SW1    ethernet14   rx  up    10Gbps 12       True
5  TAP-PORT-eth16            FILTER-SW1    ethernet16   rx  up    10Gbps 13       True
6  TAP-PORT-eth18            FILTER-SW1    ethernet18   rx  up    10Gbps 14       True
7  TAP-PORT-eth20            FILTER-SW1    ethernet20   rx  up    10Gbps 16       True
8  TAP-PORT-eth22            FILTER-SW1    ethernet22   rx  up    10Gbps 17       True
```

12.3 Auto VLAN Mode

Analysis tools often use VLAN tags to identify the filter interface receiving traffic. How VLAN IDs are assigned to traffic depends on which auto-VLAN mode is enabled. The system automatically assigns the VLAN ID from a configurable range of VLAN IDs from ***1*** to ***4094*** by default. Available auto-VLAN modes behave as follows:

- Push per Policy (default): Automatically adds a unique VLAN ID to all traffic selected by a specific policy. This setting enables tag-based forwarding.
- Push per Filter: Automatically adds a unique VLAN ID from the default auto-vlan-range (1-4094) to each filter interface. A new vlan range can be specified using the auto-vlan-range command. Manually assign any VLAN ID not in the auto-VLAN range to a filter interface.

The following table summarizes how VLAN tagging occurs with the different Auto VLAN modes.

Traffic with VLAN tag type	**push-per-policy Mode** **(Applies to all supported switches)**	**push-per-filter Mode** **(Applies to all supported switches)**
Untagged	Pushes a single tag	Pushes a single tag
Single tag	Pushes an outer (second) tag	Pushes an outer (second) tag
Two tags	Pushes an outer (third) tag. Except on T3-based switches, it rewrites the outer tag. Due to this outer customer VLAN is replaced by DMF policy VLAN.	Rewrites the outer tag. Due to this outer customer VLAN is replaced by DMF filter VLAN.

Note: When enabling push-per-policy, the **`auto-delivery-interface-vlan-strip`** feature is enabled (if disabled) before enabling push-per-policy. When enabling push-per-filter, the global delivery strip option is not enabled if previously disabled.

The following table summarizes how different Auto VLAN modes affect supported applications and services.

Note: Matching on untagged packets cannot be applied to DMF policies when in push-per-policy mode.

Auto-VLAN Mode	**Supported Platform**	**TCAM Optimization in the Core**	**L2 GRE Tunnels Support**	**Q-in-Q Packets Preserve Both Original Tags**	**Supported DMF Service Node Services**	**Manual Tag to Filter Interface**
Push per Policy (default)	All	Yes	Yes	Yes	All	Policy tag overwrites manual
Push per Filter	All	No	Yes	No	All	Configuration not allowed

Tag-based forwarding, which improves traffic forwarding and reduces TCAM utilization on the monitoring fabric switches, is only enabled when choosing the push-per-policy option.

Use the CLI or the GUI to configure Auto VLAN Mode as described in the following topics.

- Configuring Auto VLAN Mode using the CLI
- Configuring Auto VLAN Mode using the GUI

12.3.1 Configuring Auto VLAN Mode using the CLI

To set the auto VLAN mode, perform the following steps:

1. When setting the auto VLAN mode to push-per-filter, define the range of automatically assigned VLAN IDs by entering the following command from config mode:

```
auto-vlan-range vlan-min start vlan-max end
```

Replace ***start*** and ***end*** with the first and last VLAN ID in the range. For example, the following command assigns a range of 100 VLAN IDs from ***3994*** to ***4094***:

```
controller-1(config)# auto-vlan-range vlan-min 3994 vlan-max 4094
```

2. Select the VLAN mode using the following command from config mode:

```
auto-vlan-mode command { push-per-filter | push-per-policy }
```

Find details of the impact of these options in the Managing VLAN Tags in the Monitoring Fabric section.

For example, the following command adds a unique outer VLAN tag to each packet received on each filter interface:

```
controller-1(config)# auto-vlan-mode push-per-filter
Switching to auto vlan mode would cause policies to be re-installed. Enter "yes" (or "y")
to continue: y
```

3. To display the configured VLAN mode, enter the **`show fabric`** command, as in the following example:

```
controller-1# show fabric
~~~~~~~~~~~~~~~~~~~~~ Aggregate Network State ~~~~~~~~~~~~~~~~~~~~~
Number of switches                          : 5
Inport masking                              : True
Start time                                  : 2018-11-02 23:42:29.183000 UTC
Number of unmanaged services                : 0
Filter efficiency                           : 1:1
Number of switches with service interfaces  : 0
Total delivery traffic (bps)                : 411Kbps
Number of managed service instances         : 0
Number of service interfaces                : 0
Match mode                                  : full-match
Number of delivery interfaces               : 13
Max pre-service BW (bps)                    : -
Auto VLAN mode                              : push-per-filter
Number of switches with delivery interfaces : 4
Number of managed devices                   : 1
Uptime                                      : 2 days, 19 hours
Total ingress traffic (bps)                 : 550Kbps
Max overlap policies (0=disable)            : 10
Auto Delivery Interface Strip VLAN          : False
Number of core interfaces                   : 219
Max filter BW (bps)                         : 184Gbps
Number of switches with filter interfaces   : 5
State                                       : Enabled
Max delivery BW (bps)                       : 53Gbps
Total pre-service traffic (bps)             : -
Track hosts                                 : True
Number of filter interfaces                 : 23
Number of active policies                   : 3
Number of policies                          : 25
------------------------output truncated------------------------
```

4. To display the VLAN IDs assigned to each policy, enter the **`show policy`** command, as in the following example:

```
controller-1> show policy
# Policy Name Action Runtime Status Type Priority Overlap Priority
‹→Push VLAN Filter BW Delivery BW Post Match Filter Traffic Delivery Traffic Services
-|-------------------------|-------|-------------|----------|--------|----------------|--
‹→-------|---------|-----------|------------------------|----------------|----------------
‹→-----|
1 GENERATE-NETFLOW-RECORDS forward installed Configured 100 0 4
‹→ 100Gbps 10Gbps - - DMF-OOB-NETFLOWSERVICE
2 P1 forward inactive Configured 100 0 1
‹→ - 1Gbps - -
3 P2 forward inactive Configured 100 0 3
‹→ - 10Gbps - -
4 TAP-WINDOWS10-NETWORK forward inactive Configured 100 0 2
‹→ 21Gbps 1Gbps - -
5 TIMESTAMP-INCOMING-PACKETS forward inactive Configured 100 0 5
‹→ - 100Gbps - - DMF-OOB-TIMESTAMPINGSERVICE
controller -1>)#
```



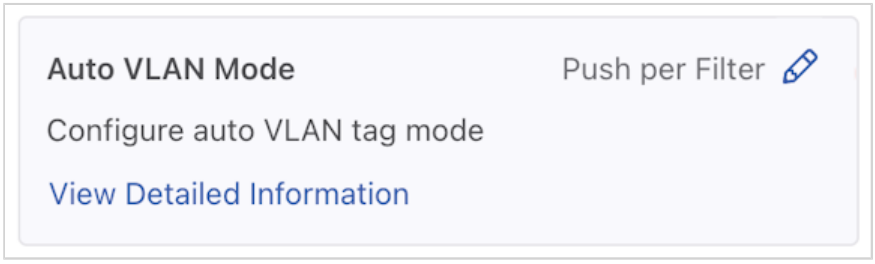
Note: The strip VLAN option, when enabled, removes the outer VLAN tag, including the VLAN ID applied by any rewrite VLAN option.

12.3.2 Configuring Auto VLAN Mode using the GUI

Auto VLAN Mode

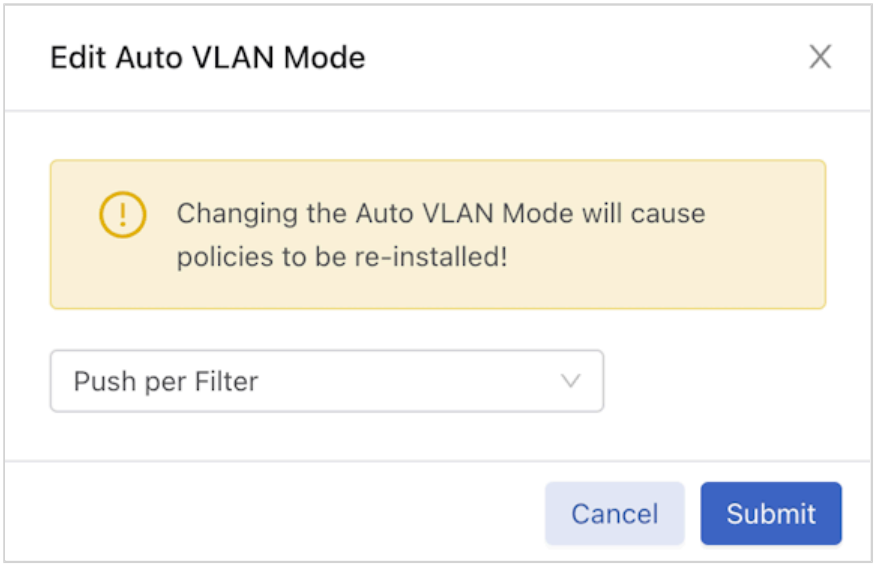
1. Control the configuration of this feature using the **Edit** icon by locating the corresponding card and the **pencil** icon.

   Figure 12-4: Auto VLAN Mode Config



2. A confirmation edit dialogue window appears, displaying the corresponding prompt message.

   Figure 12-5: Edit VLAN Mode



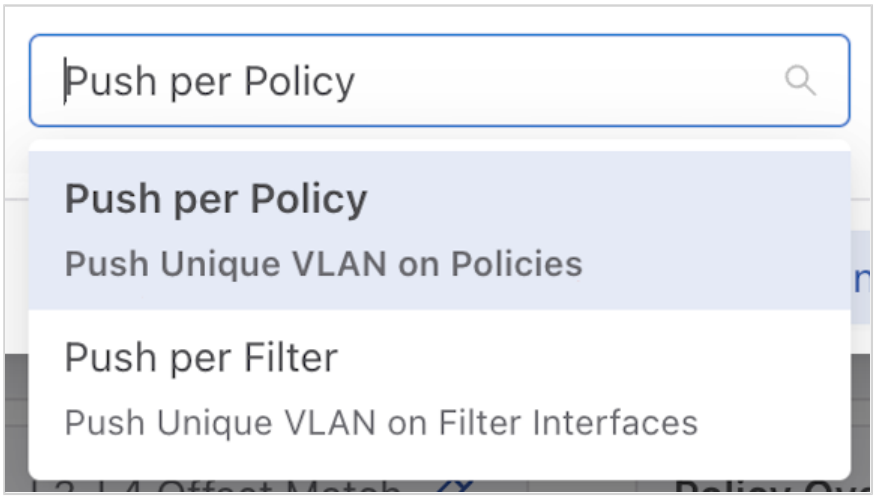
3. To configure different modes, select the drop-down arrow to open the menu.

   Figure 12-6: Drop-down Example



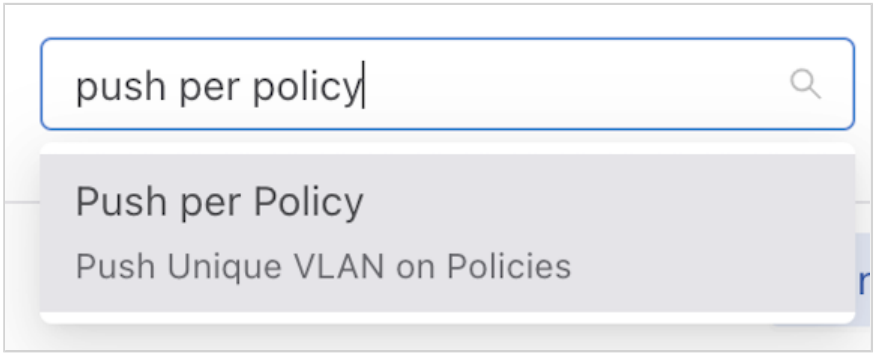
4. From the drop-down menu, select the desired mode.

Figure 12-7: Push Per Policy



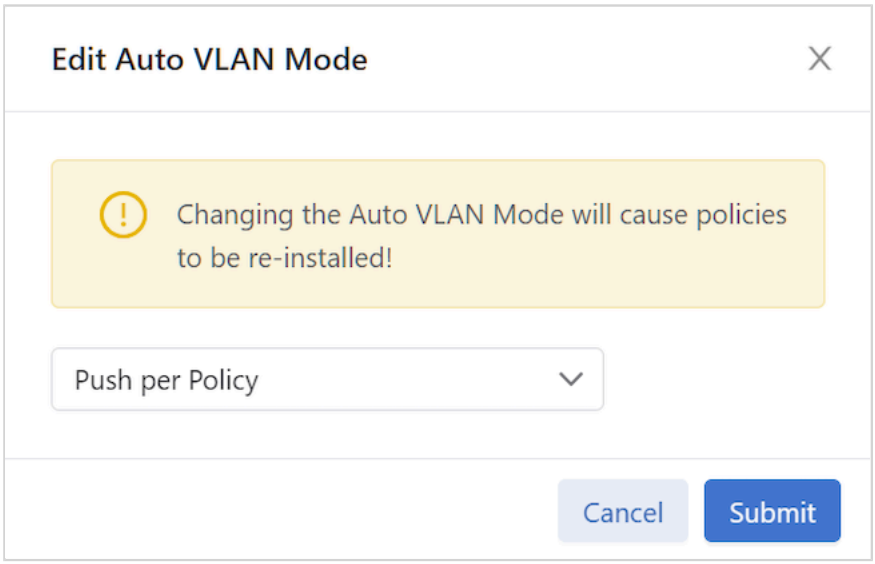
5. Alternatively, enter the desired mode name in the input area.

Figure 12-8: Push Per Policy



6. Use **Submit** to confirm the configuration changes or **Cancel** to discard the changes.

Figure 12-9: Submit Button



7. After successfully setting the configuration, the current configuration status displays next to the **edit** icon.

Figure 12-10: Current Configuration Status



The following feature sets work in the same manner as the **Auto VLAN Mode** feature described above.

- **Device Deployment Mode**
- **Match Mode**

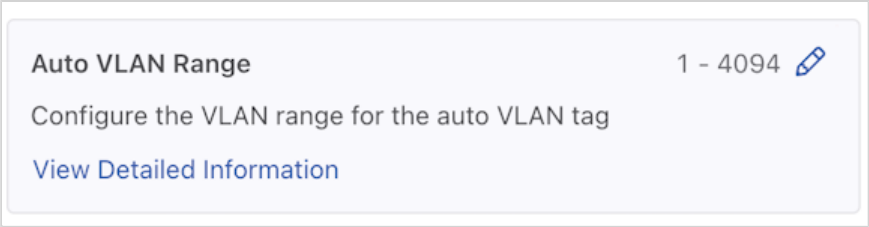
12.4 Auto VLAN Range

Auto VLAN Range

The range of automatically generated VLANs only applies when setting Auto VLAN Mode to push-per-filter. VLANs are picked from the range 1 - 4094 when not specified.

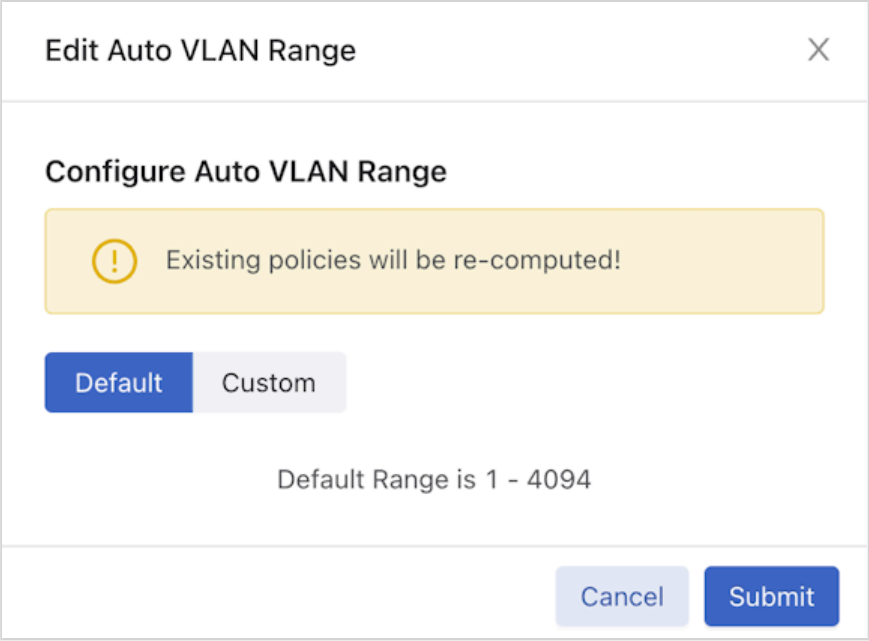
1. Control the configuration of this feature using the **Edit** icon by locating the corresponding card and selecting the **pencil** icon.

   Figure 12-11: Edit Auto VLAN Range



2. A configuration edit dialogue window pops up, displaying the corresponding prompt message. The Auto VLAN Range defaults to 1 - 4094.

   Figure 12-12: Edit Auto VLAN Range



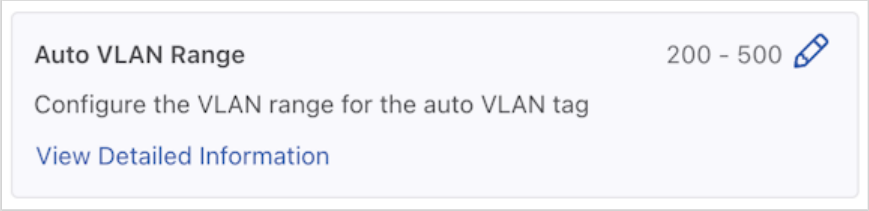
3. Select **Custom** to configure the custom range.

   Figure 12-13: Custom Button



4. Adjust range value (minimum value: 1, maximum value: 4094). There are three ways to adjust the value of a range:
   - Directly enter the desired value in the input area, with the left side representing the minimum value of the range and the right side representing the maximum value.
   - Adjust the value by dragging the **slider** using a mouse. The left knob represents the minimum value of the range, while the right knob represents the maximum value.
   - Use the up and down arrow buttons in the input area to adjust the value accordingly. Pressing the up arrow increments the value by 1, while pressing the down arrow decrements it by 1.
5. Use **Submit** to confirm the configuration changes or **Cancel** to discard the changes.
6. After successfully setting the configuration, the current configuration status displays next to the **edit** icon.

   Figure 12-14: Configuration Change Success



Configuring Auto VLAN Range using the CLI

To set the Auto VLAN Range, use the following command:

```
auto-vlan-range vlan-min start vlan-max end
```

To set the Auto VLAN Range, replace ***start*** and ***end*** with the first and last VLAN ID in the desired range.

For example, the following command assigns a range of 100 VLAN IDs from 3994 to 4094:

```
controller-1(config)# auto-vlan-range vlan-min 3994 vlan-max 4094
```

12.5 Auto VLAN Strip

The strip VLAN option removes the outer VLAN tag before forwarding the packet to a delivery interface. Only the outer tag is removed if the packet has two VLAN tags. If it has no VLAN tag, the packet is not modified. Users can remove the VLAN ID on traffic forwarded to a specific delivery interface globally for all delivery interfaces. The strip VLAN option removes any VLAN ID applied by the rewrite VLAN option.

The strip vlan option removes the VLAN ID on traffic forwarded to the delivery interface. The following are the two methods available:

- Remove VLAN IDs fabric-wide for all delivery interfaces. This method removes only the VLAN tag added by DMF Fabric.
- On specific delivery interfaces. This method has four options:
  - Keep all tags intact. Preserves the VLAN tag added by DMF Fabric and other tags in the traffic using strip-no-vlan option during delivery interface configuration.
  - Remove only the outer VLAN tag the DANZ Monitoring Fabric added using the strip-one-vlan option during delivery interface configuration.
  - Remove only the second (inner) tag. Preserves the VLAN (outer) tag added by DMF Fabric and removes the second (inner) tag in the traffic using the strip-second-vlan option during delivery interface configuration.
  - Remove two tags. Removes the outer VLAN tag added by DMF fabric and inner vlan tag in the traffic using the strip-two-vlan option during delivery interface configuration.



Note: The strip vlan command for a specific delivery interface overrides the fabric-wide strip vlan option.

By default, the VLAN ID is stripped when DMF adds it to enable the following options:

- Push per Policy
- Push per Filter
- Rewrite VLAN under filter-interfaces

Tagging and stripping VLANs as they ingress and egress DMF differs depending on whether the switch is a Trident 3-based.

Use the CLI or the GUI to configure Auto VLAN Strip as described in the following topics.

- Auto VLAN Strip using the CLI
- Auto VLAN Strip using the GUI

12.5.1 Auto VLAN Strip using the CLI

The strip VLAN option removes the outer VLAN tag before forwarding the packet to a delivery interface. Only the outer tag is removed if the packet has two VLAN tags. If it has no VLAN tag, the packet is not modified. Users can remove the VLAN ID on traffic forwarded to a specific delivery interface or globally for all delivery interfaces. The strip VLAN option removes any VLAN ID applied by the rewrite VLAN option.

The following are the two methods available:

- **Remove VLAN IDs fabric-wide for all delivery interfaces**: This method removes only the VLAN tag added by the DMF Fabric.
- **Remove VLAN IDs only on specific delivery interfaces**: This method has four options:
  - Keep all tags intact. Preserves the VLAN tag added by DMF Fabric and other tags in the traffic using **strip-no-vlan** option during delivery interface configuration.
  - Remove only the outer VLAN tag the DANZ Monitoring Fabric added using the **strip-one-vlan** option during delivery interface configuration.
  - Remove only the second (inner) tag. Preserves the VLAN (outer) tag added by DMF and removes the second (inner) tag in the traffic using the **strip-second-vlan** option during delivery interface configuration.
  - Remove two tags. Removes the outer VLAN tag added by DMF fabric and the inner VLAN tag in the traffic using the **strip-two-vlan** option during delivery interface configuration.



Note: The **`strip vlan`** command for a specific delivery interface overrides the **`fabric-wide strip VLAN`** option.

By default, the VLAN ID is stripped when DMF adds it as a result of enabling the following options:

- push-per-policy
- push-per-filter
- rewrite vlan under filter-interfaces

To view the current **`auto-delivery-interface-vlan-strip`** configuration, enter the following command:

```
controller-1> show running-config feature details
! deployment-mode
deployment-mode pre-configured
! auto-delivery-interface-vlan-strip
auto-delivery-interface-vlan-strip
! auto-vlan-mode
auto-vlan-mode push-per-policy
! auto-vlan-range
auto-vlan-range vlan-min 3200 vlan-max 4094
! crc
crc
! match-mode
match-mode full-match
! tunneling
tunneling
! allow-custom-priority
allow-custom-priority
! inport-mask
no inport-mask
```

```
! overlap-limit-strict
no overlap-limit-strict
! overlap-policy-limit
overlap-policy-limit 10
! packet-capture
packet-capture retention-days 7
```

To view the current **auto-delivery-interface-vlan-strip** state, enter the following command:

```
controller-1> show fabric
~~~~~~~~~~~~~~~~~~~~ Aggregate Network State ~~~~~~~~~~~~~~~~~~~~
Number of switches : 5
Inport masking : True
Start time : 2018-10-16 22:30:03.345000 UTC
Number of unmanaged services : 0
Filter efficiency : 3005:1
Number of switches with service interfaces : 0
Total delivery traffic (bps) : 232bps
Number of managed service instances : 0
Number of service interfaces : 0
Match mode : l3-l4-match
Number of delivery interfaces : 24
Max pre-service BW (bps) : -
Auto VLAN mode : push-per-policy
Number of switches with delivery interfaces : 5
Number of managed devices : 1
Uptime : 21 hours, 53 minutes
Total ingress traffic (bps) : 697Kbps
Max overlap policies (0=disable) : 10
Auto Delivery Interface Strip VLAN : True
```

To disable this global command, enter the following command:

```
controller-1(config-switch-if)# no auto-delivery-interface-vlan-strip
```

The delivery interface level command to strip the VLAN overrides the global **auto-delivery-interface-vlan-strip** command. For example, when global VLAN stripping is disabled or to override the default strip option on a delivery interface use the below options:

To strip the VLAN added by DMF fabric on a specific delivery interface, use the following command:

```
controller-1(config-switch-if)# role delivery interface-name TOOL-PORT-1 strip-one-vlan
```

When global VLAN stripping is enabled, it strips only the outer VLAN ID. To remove outer VLAN ID that was added by DMF as well as the inner VLAN ID, enter the following command:

```
controller-1(config-switch-if)# role delivery interface-name TOOL-PORT-1 strip-two-vlan
```

To strip only the inner VLAN ID and preserve the outer VLAN ID that DMF added, use the following command:

```
controller-1(config-switch-if)# role delivery interface-name TOOL-PORT-1 strip-second-vlan
```

To preserve the VLAN tag added by DMF and other tags in the traffic, use the following command:

```
controller-1(config-switch-if)# role delivery interface-name TOOL-PORT-1 strip-no-vlan
```

Note: For all modes VLAN ID stripping is supported at both the global and delivery interface levels. The **rewrite-per-policy** and **rewrite-per-filter** options have been removed in ***DMF Release 6.0*** because the **push-per-policy** and **push-per-filter** options now support the related use cases.

The syntax for the strip VLAN ID feature is as follows:

```
controller-1(config-switch-if)# role delivery interface-name name [strip-no-vlan | strip-onevlan |
 strip-second-vlan | strip-two-vlan]
```

Use the option to leave all VLAN tags intact, remove the outermost tag, remove the second (inner) tag, or remove the outermost two tags, as required.

By default, VLAN stripping is enabled and the outer VLAN added by DMF is removed.

To preserve the outer VLAN tag, enter the **`strip-no-vlan`** command, as in the following example, which preserves the outer VLAN ID for traffic forwarded to the delivery interface ***TOOL-PORT-1***:

```
controller-1(config-switch-if)# role delivery interface-name TOOL-PORT-1 strip-no-vlan
```

When global VLAN stripping is disabled, the following commands remove the outer VLAN tag, added by DMF, on packets transmitted to the specific delivery interface ***ethernet20*** on ***DMF-DELIVERY-SWITCH-1***:

```
controller-1(config)# switch DMF-DELIVERY-SWITCH-1
controller-1(config-switch)# interface ethernet20
controller-1(config-switch-if)# role delivery interface-name TOOL-PORT-1 strip-one-vlan
```

To restore the default configuration, which is to strip the VLAN IDs from traffic to every delivery interface, enter the following command:

```
controller-1(config)# auto-delivery-interface-vlan-strip
This would enable auto delivery interface strip VLAN feature.
Existing policies will be re-computed. Enter "yes" (or "y") to continue: yes
```

As mentioned earlier, tagging and stripping VLANs as they ingress and egress DMF differs based on whether the switch uses a Trident 3 chipset. The following scenarios show how DMF behaves in different VLAN modes with various knobs set.

Scenario 1

- VLAN mode: ***Push per Policy***
- Filter interface on any switch except a Trident 3 switch
- Delivery interface on any switch
- Global VLAN stripping is **enabled**

Table 9: Behavior of Traffic as It Egresses with Different Strip Options on a Delivery Interface

VLAN tag type	No Configuration	strip-no-VLAN	strip-one-VLAN	strip-second-VLAN	strip-two-VLAN
	DMF policy VLAN is stripped automatically on delivery inter- face using default global strip VLAN added by DMF	DMF policy VLAN and customer VLAN preserved	Strips the outermost VLAN that is DMF policy VLAN	DMF policy VLAN is preserved and outermost customer VLAN is removed	Strip two VLANs, DMF policy VLAN and customer outer VLAN removed
Untagged	Packets exit DMF as untagged packets	Packets exit DMF as singly tagged packets. VLAN in the packet is DMF policy VLAN.	Packets exit DMF as untagged packets.	Packets exit DMF as single-tagged traffic. VLAN in the packet is DMF policy VLAN.	Packets exit DMF as untagged traffic.
Singly Tagged	Packets exit DMF as single-tagged traffic with customer VLAN.	Packets exit DMF as doubly tagged packets. Outer VLAN in the packet is DMF policy VLAN.	Packets exit DMF as single-tagged traffic with customer VLAN.	Packets exit DMF as single-tagged traffic. VLAN in the packet is DMF policy VLAN.	Packets exit DMF as untagged traffic.
Doubly Tagged	Packet exits DMF as doubly tagged traffic. Both VLANs are customer VLANs.	Packet exits DMF as triple-tagged packets. Outermost VLAN in the packet is the DMF policy VLAN.	Packet exits DMF as doubly tagged traffic. Both VLANs are customer VLANs.	Packet exits DMF as double-tagged packets. Outer VLAN is DMF policy VLAN, inner VLAN is inner customer VLAN in the original packet.	Packet exits DMF as singly tagged traffic. VLAN in the packet is the inner customer VLAN.

Scenario 2

- VLAN Mode: ***Push per Policy***
- Filter interface on any switch except a Trident 3 switch
- Delivery interface on any switch
- Global VLAN strip is **disabled**

Table 10: Behavior of Traffic as It Egresses with Different Strip Options on a Delivery Interface

VLAN tag type	No Configuration	strip-no-VLAN	strip-one-VLAN	strip-second-VLAN	strip-two-VLANs
	DMF policy VLAN and customer VLAN are preserved	DMF policy VLAN and customer VLAN are preserved	Strips only the outermost VLAN that is DMF policy VLAN	DMF policy VLAN is preserved and outer most customer VLAN is removed	Strip two VLANs, DMF policy VLAN and customer outer VLAN removed
Untagged	Packet exits DMF as singly tagged packets. VLAN in the packet is DMF policy VLAN.	Packet exits DMF as singly tagged packets. VLAN in the packet is DMF policy VLAN.	Packet exits DMF as untagged packets.	Packet exits DMF as single-tagged traffic. VLAN in the packet is DMF policy VLAN.	Packet exits DMF as untagged traffic.
Singly Tagged	Packet exits DMF as doubly tagged packets. Outer VLAN in packet is DMF policy VLAN and inner VLAN is customer outer VLAN.	Packet exits DMF as doubly tagged packets. Outer VLAN in the packet is DMF policy VLAN.	Packet exits DMF as single-tagged traffic with customer VLAN.	Packet exits DMF as single-tagged traffic. VLAN in the packet is DMF policy VLAN.	Packets exits DMF as untagged traffic.
Doubly Tagged	Packet exits DMF as triple-tagged packets. Outermost VLAN in the packet is the DMF policy VLAN.	Packet exits DMF as triple-tagged packets. Outermost VLAN in the packet is the DMF policy VLAN.	Packet exits DMF as doubly tagged traffic. Both VLANs are customer VLANs.	Packet exits DMF as doubly tagged packets. Outer VLAN is DMF policy VLAN, inner VLAN is inner customer VLAN in the original packet.	Packet exits DMF as singly tagged traffic. VLAN in the packets is the inner customer VLAN.

Scenario 3

- VLAN Mode - ***Push per Policy***
- Filter interface on a Trident 3 switch
- Delivery interface on any switch
- Global VLAN strip is **enabled**

Table 11: Behavior of traffic as it egresses with different strip options on a delivery interface

VLAN tag type	No Configuration	strip-no-VLAN	strip-one-VLAN	strip-second-VLAN	strip-two-VLAN
	DMF policy VLAN is stripped automatically on delivery interface using default global strip VLAN added by DMF	DMF policy VLAN and customer VLAN preserved	Strips the outermost VLAN that is DMF policy VLAN	DMF policy VLAN is preserved and outermost customer VLAN is removed	Strip two VLANs , DMF policy VLAN and customer outer VLAN removed
Untagged	Packet exits DMF as untagged packets.	Packet exits DMF as singly tagged packets. VLAN in the packet is DMF policy VLAN.	Packet exits DMF as untagged packets.	Packet exits DMF as single-tagged traffic. VLAN in the packet is DMF policy VLAN.	Packet exits DMF as untagged traffic.
Singly Tagged	Packet exits DMF as single-tagged traffic with customer VLAN.	Packet exits DMF as doubly tagged packets. Outer VLAN in the packet is DMF policy VLAN.	Packet exits DMF as single tagged traffic with customer VLAN.	Packet exits DMF as single-tagged traffic. VLAN in the packet is DMF policy VLAN.	Packet exits DMF as untagged traffic.
Doubly Tagged	Packet exits DMF as singly tagged traffic. VLAN in the packet is the inner customer VLAN	Packet exits DMF as doubly tagged traffic. Outer customer VLAN is replaced by DMF policy VLAN.	Packet exits DMF as singly tagged traffic. VLAN in the packet is the inner customer VLAN.	Packet exits DMF as singly tagged traffic. VLAN in the packet is the DMF policy VLAN.	Packet exits DMF as untagged traffic.

Scenario 4

- VLAN Mode - ***Push per Filter***
- Filter interface on any switch
- Delivery interface on any switch
- Global VLAN strip is **enabled**

Table 12: Behavior of Traffic as It Egresses with Different Strip Options on a Delivery Interface

VLAN tag type	No Configuration	strip-no-VLAN	strip-one-VLAN	strip-second-VLAN	strip-two-VLAN
	DMF filter VLAN is stripped automatically on delivery interface using global strip VLAN added by DMF.	DMF filter VLAN and customer VLAN preserved.	Strips the outermost VLAN that is DMF filter VLAN.	DMF filter VLAN is preserved and outermost customer VLAN is removed.	Strip two VLANs, DMF filter interface VLAN and customer outer VLAN removed.
Untagged	Packet exits DMF as untagged packets.	Packet exits DMF as singly tagged packets. VLAN in the packet is DMF filter interface VLAN.	Packet exits DMF as untagged packets.	Packet exits DMF as single tagged traffic. VLAN in the packet is DMF filter inter- face VLAN.	Packet exits DMF as untagged traffic.
Singly Tagged	Packet exits DMF as singly tagged traffic. VLAN in the packet is the customer VLAN.	Packet exits DMF as doubly tagged packets. Outer VLAN in the packet is DMF filter interface VLAN.	Packet exits DMF as singly tagged traffic. VLAN in the packet is the customer VLAN.	Packet exits DMF as singly tagged traffic. VLAN in the packet is DMF filter interface VLAN.	Packet exits DMF as untagged traffic.
Doubly Tagged	Packet exits DMF as singly tagged traffic. VLAN in the policy is the inner customer VLAN.	Packet exits DMF as doubly tagged traffic. Outer customer VLAN is replaced by DMF filter interface VLAN.	Packet exits DMF as singly tagged traffic. VLAN in the policy is the inner customer VLAN.	Packet exits DMF as singly tagged traffic. VLAN in the policy is the DMF filter interface VLAN.	Packet exits DMF as untagged traffic.

Scenario 5

- VLAN Mode - ***Push per Filter***
- Filter interface on any switch
- Delivery interface on any switch
- Global VLAN strip is **disabled**

Table 13: Behavior of Traffic as It Egresses with Different Strip Options on a Delivery Interface

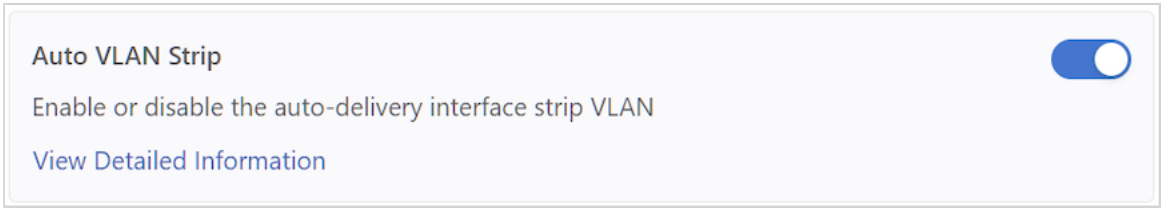
VLAN tag type	No Configuration	strip-no-VLAN	strip-one-VLAN	strip-second-VLAN	strip-two-VLAN
	DMF filter VLAN is stripped automatically on delivery interface using global strip VLAN added by DMF.	DMF filter VLAN and customer VLAN preserved.	Strips the outermost VLAN that is DMF filter VLAN.	DMF filter VLAN is preserved and outermost customer VLAN is removed.	Strip two VLANs, DMF filter interface VLAN and customer outer VLAN removed.
Untagged	Packet exits DMF as singly tagged packets. VLAN in the packet is DMF filter interface VLAN.	Packet exits DMF as singly tagged packets. VLAN in the packet is DMF filter interface VLAN.	Packet exits DMF as untagged packets.	Packet exits DMF as singly tagged traffic. VLAN in the packet is DMF filter interface VLAN.	Packet exits DMF as untagged traffic.
Singly Tagged	Packet exits DMF as doubly tagged traffic. Outer VLAN in the packet is DMF filter VLAN and inner VLAN is the customer VLAN.	Packet exits DMF as doubly tagged packets. Outer VLAN in the packet is DMF filter interface VLAN.	Packet exits DMF as single tagged traffic. VLAN in the packet is the customer VLAN.	Packet exits DMF as singly tagged traffic. VLAN in the packet is DMF filter interface VLAN.	Packet exits DMF as untagged traffic.
Doubly Tagged	Packet exits DMF as doubly tagged traffic. Outer customer VLAN is replaced by DMF filter interface VLAN.	Packet exits DMF as doubly tagged traffic. Outer customer VLAN is replaced by DMF filter interface VLAN.	Packet exits DMF as singly tagged traffic. VLAN in the policy is the inner customer VLAN.	Packet exits DMF as singly tagged traffic. VLAN in the policy is the DMF filter interface VLAN.	Packet exits DMF as untagged traffic.

12.5.2 Auto VLAN Strip using the GUI

Auto VLAN Strip

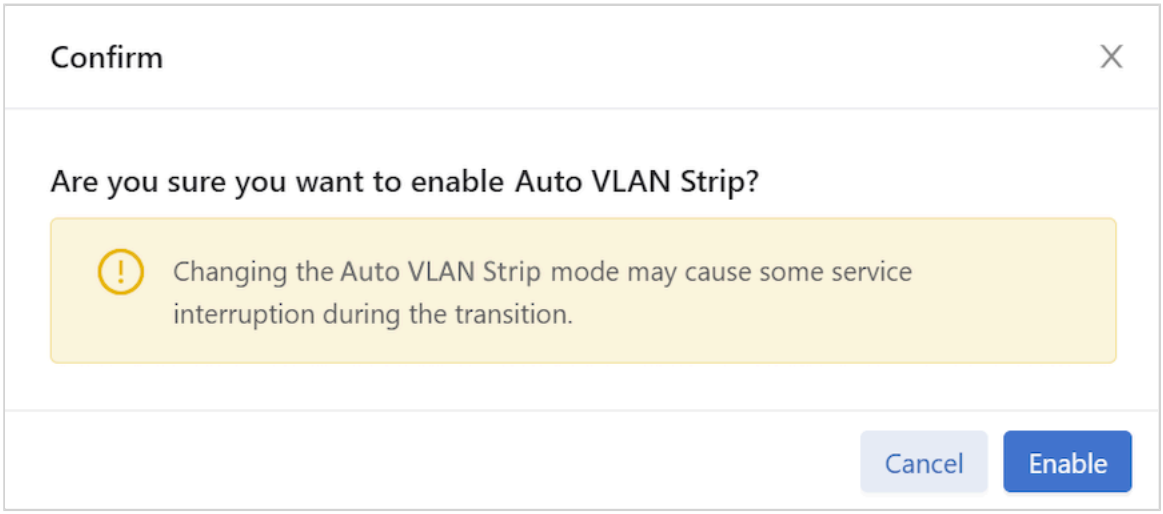
1. A **toggle button** controls the configuration of this feature. Locate the corresponding card and use the **toggle** switch.

   Figure 12-15: Toggle Switch



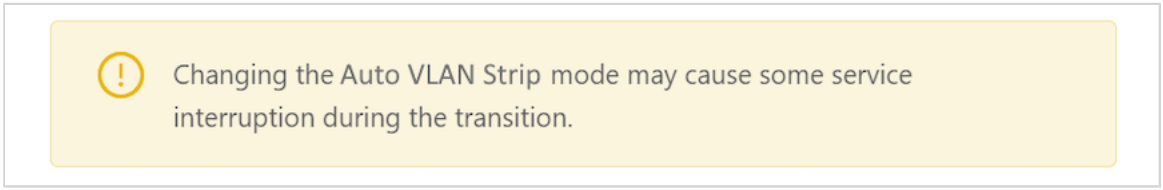
2. A confirm window pops up, displaying the corresponding prompt message. Use **Enable** to confirm the configuration changes or **Cancel** to cancel the configuration. Conversely, to disable the configuration, select **Disable**.

   Figure 12-16: Confirm / Enable



3. Review any warning messages that appear in the confirmation window during the configuration process.

   Figure 12-17: Warning Message - Changing



The following feature sets work in the same manner as the **Auto VLAN Strip** feature described above.

- **CRC Check**
- **Custom Priority**
- **Inport Mask**
- **Policy Overlap Limit Strict**
- **Retain User Policy VLAN**
- **Tunneling**

12.6 CRC Check

If the Switch CRC option is enabled, which is the default, each DMF switch drops incoming packets that enter the fabric with a CRC error. The switch generates a new CRC if the incoming packet was modified using an option that modifies the original CRC checksum, which includes the push VLAN, rewrite VLAN, strip VLAN, and L2 GRE tunnel options.



Note: Enable the Switch CRC option to use the DMF tunneling feature.

If the Switch CRC option is disabled, DMF switches do not check the CRC of incoming packets and do not drop packets with CRC errors. Also, switches do not generate a new CRC if the packet is modified. This mode is helpful if packets with CRC errors need to be delivered to a destination tool unmodified for analysis. When disabling the Switch CRC option, ensure the destination tool does not drop packets having CRC errors. Also, recognize that CRC errors will be caused by modification of packets by DMF options so that these CRC errors are not mistaken for CRC errors from the traffic source.



Note: When the Switch CRC option is disabled, packets going to the Service Node or Recorder Node are dropped because a new CRC is not calculated when push-per-policy or push-per-filter adds a VLAN tag.

Enable and disable CRC Check using the steps described in the following topics.

- CRC Check using the CLI
- CRC Check using the GUI

12.6.1 CRC Check using the CLI

If the Switch CRC option is enabled, which is the default, each DMF switch drops incoming packets that enter the fabric with a CRC error. The switch generates a new CRC if the incoming packet was modified using one option that modifies the original CRC checksum, which includes the push VLAN, rewrite VLAN, strip VLAN, and L2 GRE tunnel options.



Note: Enable the Switch CRC option to use the DMF tunneling feature.

If the Switch CRC option is disabled, DMF switches do not check the CRC of incoming packets and do not drop packets with CRC errors. Also, switches do not generate a new CRC if the packet is modified. This mode is helpful if packets with CRC errors need to be delivered to a destination tool unmodified for analysis. When disabling the Switch CRC option, ensure the destination tool does not drop packets having CRC errors. Also, recognize that CRC errors will be caused by modification of packets by DMF options so that these CRC errors are not mistaken for CRC errors from the traffic source.

To disable the Switch CRC option, enter the following command from config mode:

```
controller-1(config)# no crc
Disabling CRC mode may cause problems to tunnel interface. Enter "yes" (or "y") to continue: y
```

In the event the Switch CRC option is disabled, re-enable the Switch CRC option using the following command from config mode:

```
controller-1(config)# crc
Enabling CRC mode would cause packets with crc error dropped. Enter "yes" (or "y
") to continue: y
```



Tip: To enable or disable the CRC through the GUI, refer to the chapter, Check CRC using the GUI.



Note: When the Switch CRC option is disabled, packets going to the service node or recorder node are dropped because a new CRC is not calculated when **push-per-policy** or **push-per-filter** adds a VLAN tag.

12.6.2 CRC Check using the GUI

From the DMF Features page, proceed to the CRC Check feature card and perform the following steps to enable the feature.

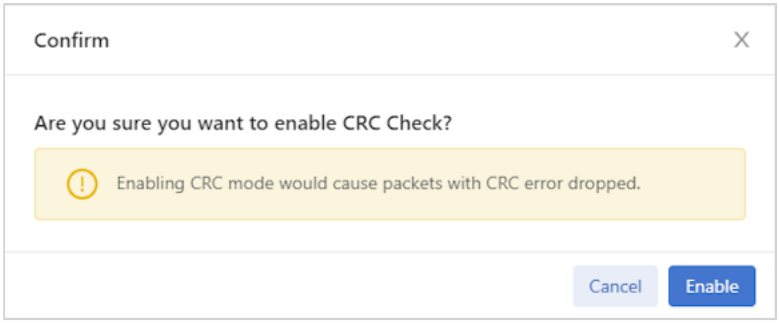
1. Select the **CRC Check** card.

   Figure 12-18: CRC Check Disabled



2. Toggle the CRC Check to **On**.
3. Confirm the activation by selecting **Enable**. Or, **Cancel** to return to the DMF Features page.

   Figure 12-19: Enable CRC Check



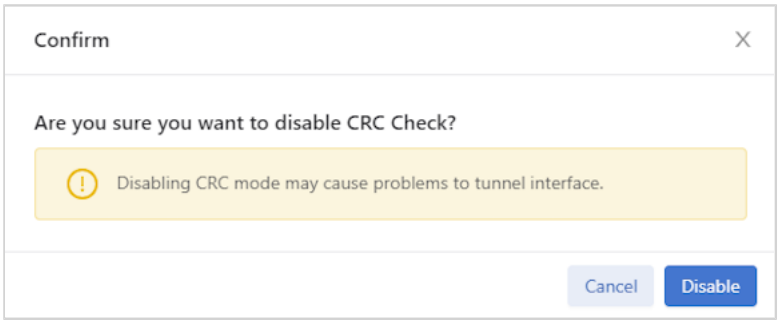
4. CRC Check is running.

   Figure 12-20: CRC Check Enabled



5. To disable the feature, toggle the CRC Check to **Off**. Select **Disable** and confirm.

   Figure 12-21: Disable CRC Check



The feature card updates with the status.

Figure 12-22: CRC Check Disabled



12.7 Custom Priority

When custom priorities are allowed, non-admin users may assign policy priorities between 0 and 100 (the default value). However, when custom priorities are not allowed, the default priority of 100 will be automatically assigned to non-admin users' policies.

Enable and disable Custom Priority using the steps described in the following topics.

- Custom Priority using the CLI
- Custom Priority using the GUI

12.7.1 Configuring Custom Priority using the GUI

From the DMF Features page, proceed to the Custom Priority feature card and perform the following steps to enable the feature.

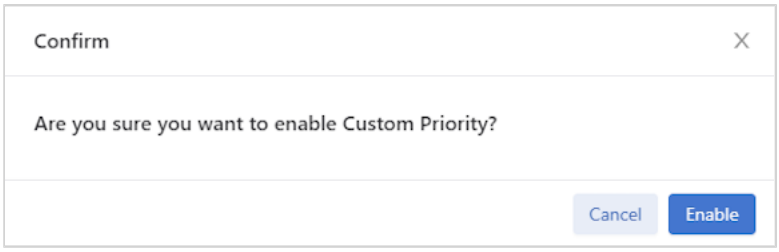
1. Select the **Custom Priority** card.

   Figure 12-23: Custom Priority Disabled



2. Toggle the Custom Priority to **On**.
3. Confirm the activation by selecting **Enable**. Or, **Cancel** to return to the DMF Features page.

   Figure 12-24: Enable Custom Priority



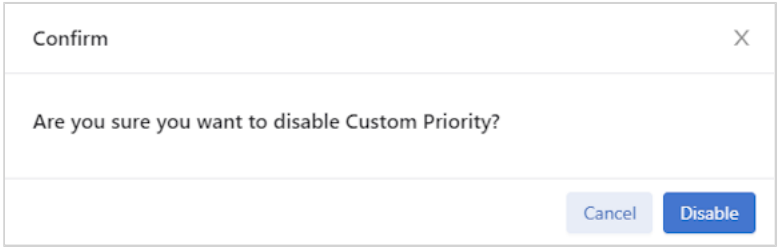
4. Custom Priority is running.

   Figure 12-25: Custom Priority Enabled



5. To disable the feature, toggle the **Custom Priority** to **Off**. Select **Disable** and confirm.

   Figure 12-26: Disable Custom Priority



The feature card updates with the status.

Figure 12-27: Custom Priority Disabled



12.7.2 Configuring Custom Priority using the CLI

To enable the Custom Priority, enter the following command:

```
controller-1(config)# allow-custom-priority
```

To disable the Custom Priority, enter the following command:

```
controller-1(config)# no allow-custom-priority
```

12.8 Device Deployment Mode

Complete the fabric switch installation in one of the following two modes:

- **Layer 2 Zero Touch Fabric (L2ZTF, Auto-discovery switch provisioning mode)**

  In this mode, switch software automatically discovers the Controller via IPv6 local link addresses and downloads and installs the appropriate Switch Light OS image from the Controller. This installation method requires all the fabric switches and the DMF Controller to be in the same Layer 2 network (IP subnet). Also, suppose the fabric switches need IPv4 addresses to communicate with SNMP or other external services. In that case, users must configure IPAM, which provides the Controller with a range of IPv4 addresses to allocate to the fabric switches.

- **Layer 3 Zero Touch Fabric (L3ZTF, Preconfigured switch provisioning mode)**

  When fabric switches are in a different Layer 2 network from the Controller, log in to each switch individually to configure network information and download the ZTF installer. Subsequently, the switch automatically downloads Switch Light OS from the Controller. This mode requires communication between the Controller and the fabric switches using IPv4 addresses, and no IPAM configuration is required.

The following table summarizes the requirements for installation using each mode:

Requirement	Layer 2 mode	Layer 3 mode
Any switch in a different subnet from the controller	No	Yes
IPAM configuration for SNMP and other IPv4 services	Yes	No
IP address assignment	IPv4 or IPv6	IPv4
Refer to this section (in User Guide)	Using L2 ZTF (Auto-Discovery) Provisioning Mode	Changing to Layer 3 (Pre-Configured) Switch Provisioning Mode

All the fabric switches in a single fabric must be installed using the same mode. If users have any fabric switches in a different IP subnet than the Controller, users must use Layer 3 mode for installing all the switches, even those in the same Layer 2 network as the Controller. Installing switches in mixed mode, with some switches using ZTF in the same Layer 2 network as the Controller, while other switches in a different subnet are installed manually or using DHCP is unsupported.

12.8.1 Configuring Device Deployment Mode using the GUI

From the DMF Features page, proceed to the Device Deployment Mode feature card and perform the following steps to manage the feature.

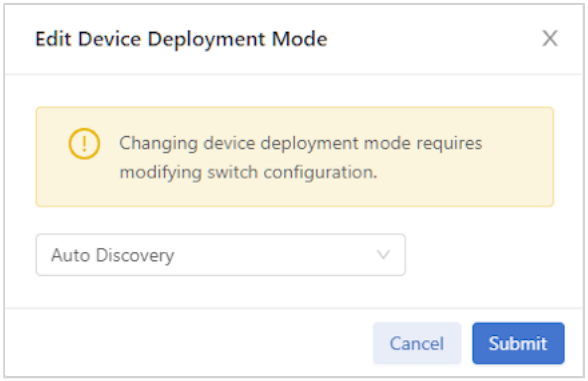
1. Select the **Device Deployment Mode** card.

   Figure 12-28: Device Deployment Mode - Auto Discovery



2. Enter the edit mode using the **pencil** icon.

   Figure 12-29: Configure Device Deployment Mode



3. Change the switching mode as required using the drop-down menu. The default mode is **Auto Discovery**.

Figure 12-30: Device Deployment Mode Options

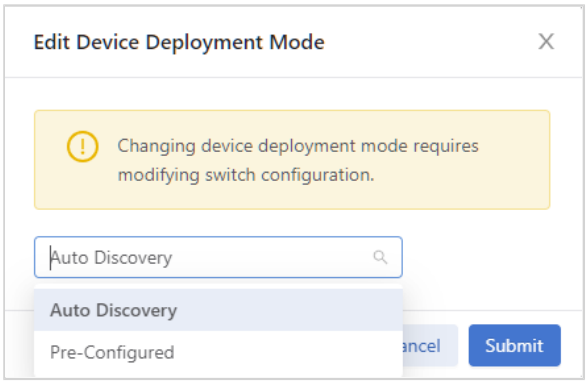
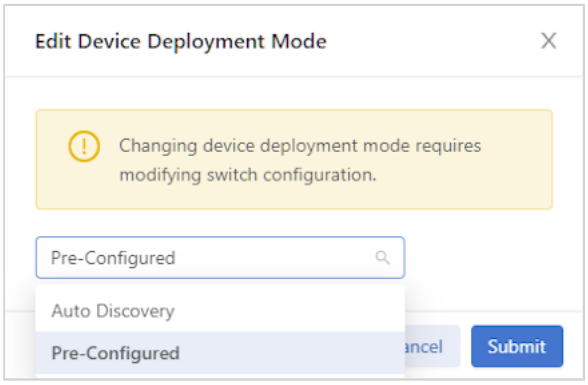


Figure 12-31: Device Deployment Mode - Pre-Configured Option



4. Select **Submit** and confirm the operation when prompted.
5. The Device Deployment Mode status updates.

Figure 12-32: Device Deployment Mode - Status Update



12.8.2 Configuring Device Deployment Mode using the CLI

Device Deployment Mode has two options: select the desired option, either auto-discovery or pre-configured, as shown below:

```
controller-1(config)# deployment-mode auto-discovery

Changing device deployment mode requires modifying switch configuration. Enter "yes" (or "y") to
 continue: y
```

```
controller-1(config)# deployment-mode pre-configured

Changing device deployment mode requires modifying switch configuration. Enter "yes" (or "y") to
 continue: y
```

12.9 Inport Mask

Enable and disable Inport Mask using the steps described in the following topics.

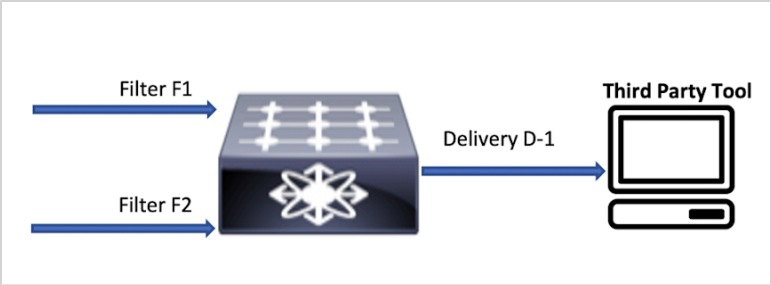
- Configure Inport Mask using the CLI
- Configure Inport Mask using the GUI

12.9.1 Inport Mask using the CLI

DANZ Monitoring Fabric implements multiple flow optimizations to reduce the number of flows programmed in the DMF switch TCAM space. This feature enables effective usage of TCAM space, and it is on by default.

When this feature is off, TCAM rules are applied for each ingress port belonging to the same policy. For example, in the following topology, if a policy was configured with ***10*** match rules and filter-interface as ***F1*** and ***F2***, then ***20*** (***10*** for ***F1*** and ***10*** for ***F2***) TCAM rows were consumed.

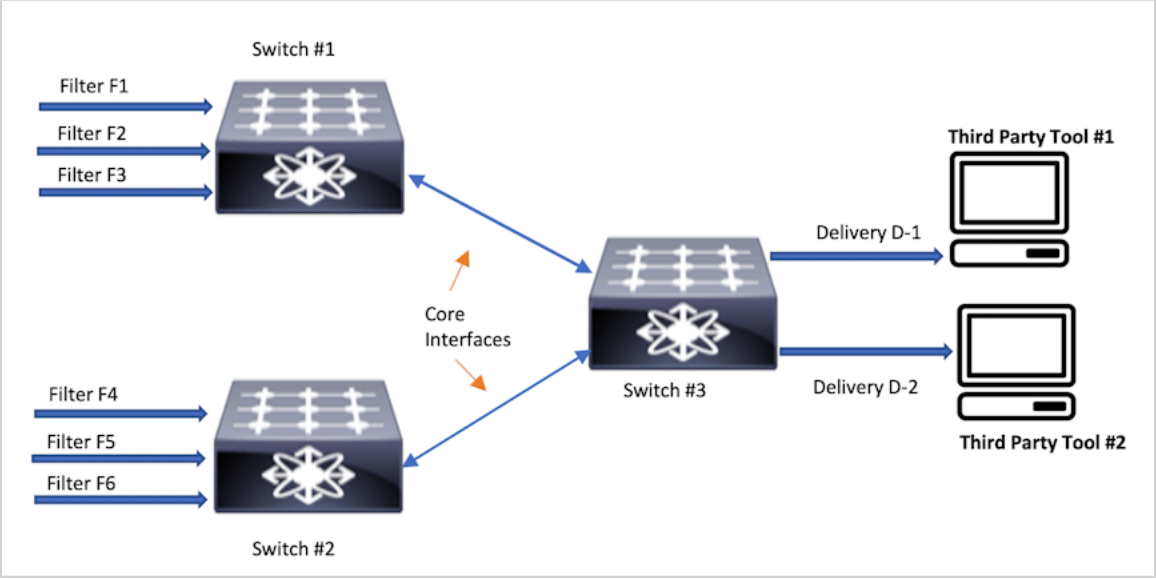
Figure 12-33: Simple Inport Mask Optimization



With inport mask optimization, only ***10*** rules are consumed. This feature optimizes TCAM usage at every level (filer, core, delivery) in the DMF network.

Consider the more complex topology illustrated below:

Figure 12-34: Complex Inport Mask Optimization



In this topology, if a policy has ***N*** rules without in-port optimization, the policy will consume ***3N*** at ***Switch 1***, ***3N*** at ***Switch 2***, and ***2N*** at ***Switch 3***. With the in-port optimization feature enabled, the policy consumes only ***N*** rules at each switch.

However, this feature loses granularity in the statistics available because there is only one set of flow mods for multiple filter ports per switch. Statistics without this feature are maintained per filter port per policy.

With inport optimization enabled, the statistics are combined for all input ports sharing rules on that switch. The option exists to obtain filter port statistics for different flow mods for each filter port. However, this requires disabling inport optimization, which is enabled by default.

To disable the inport optimization feature, enter the following command from config mode:

```
controller-1(config)# controller-1(config)# no inport-mask
```

12.9.2 Inport Mask using the GUI

From the DMF Features page, proceed to the Inport Mask feature card and perform the following steps to enable the feature.

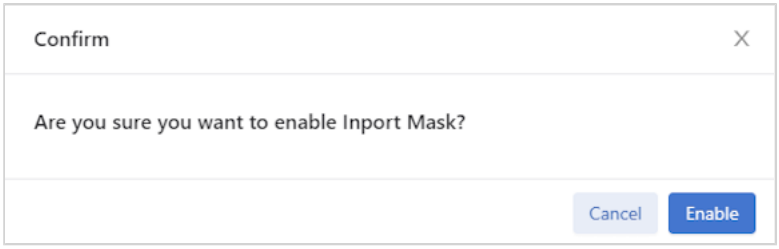
1. Select the **Inport Mask** card.

   Figure 12-35: Inport Mask Disabled



2. Toggle the Inport Mask to **On**.
3. Confirm the activation by selecting **Enable**. Or, **Cancel** to return to the DMF Features page.

   Figure 12-36: Enable Inport Mask



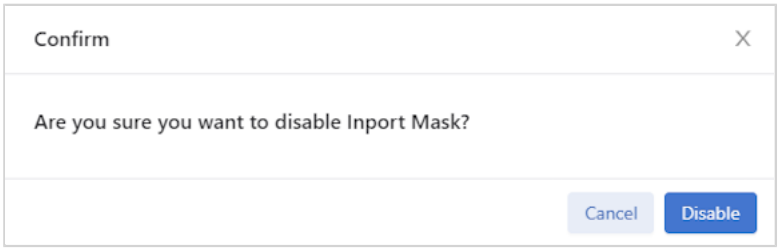
4. Inport Mask is running.

   Figure 12-37: Inport Mask Enabled



5. To disable the feature, toggle the Inport Mask to **Off**. Select **Disable** and confirm.

   Figure 12-38: Disable Inport Mask



The feature card updates with the status.

Figure 12-39: Inport Mask Disabled



12.10 Match Mode

Switches have finite hardware resources available for packet matching on aggregated traffic streams. This resource allocation is relatively static and configured in advance. The DANZ Monitoring Fabric supports three allocation schemes, referred to as switching (match) modes:

- **L3-L4 mode (default mode)**: With L3-L4 mode, fields other than **`src-mac`** and **`dst-mac`** can be used for specifying policies. If no policies use **`src-mac`** or **`dst-mac`**, the L3-L4 mode allows more match rules per switch.
- **Full-match mode**: With full-match mode, all matching fields, including **`src-mac`** and `dst-mac`, can be used while specifying policies.
- **L3-L4 Offset mode**: L3-L4 offset mode allows matching beyond the L4 header up to 128 bytes from the beginning of the packet. The number of matches per switch in this mode is the same as in full-match mode. As with L3-L4 mode, matches using **`src-mac`** and **`dst-mac`** are not permitted.



Note: Changing switching modes causes all fabric switches to disconnect and reconnect with the Controller. Also, all existing policies will be reinstalled. The switching mode applies to all DMF switches in the DANZ Monitoring Fabric. Switching between modes is possible, but any match rules incompatible with the new mode will fail.

12.10.1 Setting the Match Mode Using the CLI

To use the CLI to set the ***match*** mode, enter the following command:

```
controller-1(config)# match-mode {full-match | l3-l4-match | l3-l4-offset-match}
```

For example, the following command sets the ***match*** mode to ***full-match*** mode:

```
controller-1(config)# match-mode full-match
```

12.10.2 Setting the Match Mode Using the GUI

From the DMF Features page, proceed to the Match Mode feature card and perform the following steps to enable the feature.

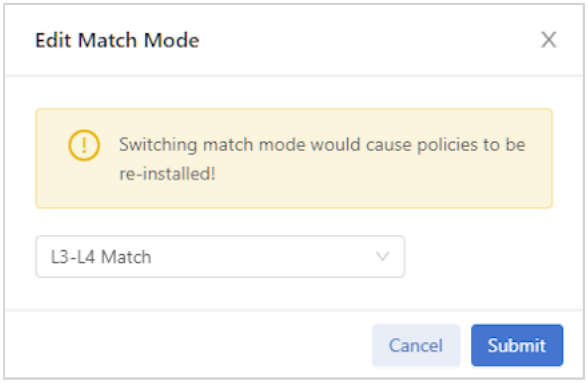
1. Select the **Match Mode** card.

   Figure 12-40: L3-L4 Match Mode



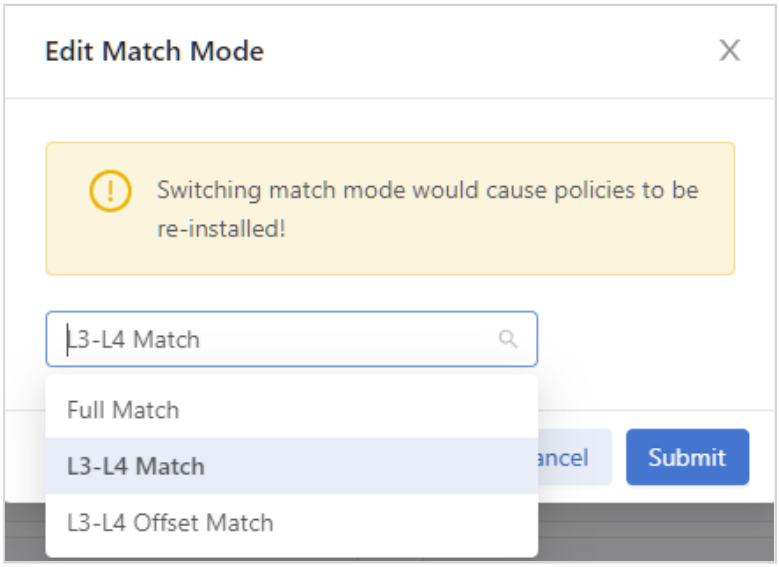
2. Enter the edit mode using the **pencil** icon.

   Figure 12-41: Configure Switching Mode



3. Change the switching mode as required using the drop-down menu. The default mode is **L3-L4 Match**.

   Figure 12-42: L3-L4 Match Options



4. Select **Submit** and confirm the operation when prompted.

Note: An error message is displayed if the existing configuration of the monitoring fabric is incompatible with the specified switching mode.

12.11 Retain User Policy VLAN

Enable and disable Retain User Policy VLAN using the steps described in the following topics.

- Retain User Policy VLAN using the CLI
- Retain User Policy VLAN using the GUI

12.11.1 Retain User Policy VLAN using the CLI

This feature will send traffic to a delivery interface with the user policy VLAN tag instead of the overlap dynamic policy VLAN tag for traffic matching the dynamic overlap policy only. This feature is supported only in ***push-per-policy*** mode. For example, policy ***P1*** with filter interface ***F1*** and delivery interface ***D1***, and policy ***P2*** with filter interface ***F1*** and delivery interface ***D2***, and overlap dynamic policy ***P1_o_P2*** is created when the overlap policy condition is met. In this case, the overlap dynamic policy is created with filter interface ***F1*** and delivery interfaces ***D1*** and ***D2***. The user policy ***P1*** assigns a VLAN (***VLAN 10***) and ***P2*** assigns a VLAN (***VLAN 20***) when it is created, and the overlap policy also assigns a VLAN (***VLAN 30***) when it is dynamically created. When this feature is enabled, traffic forwarded to ***D1*** will have a policy VLAN tag of ***P1*** (***VLAN 10***) and ***D2*** will have a policy VLAN tag of policy ***P2*** (***VLAN 20***). When this feature is disabled, traffic forwarded to ***D1*** and ***D2*** will have the dynamic overlap policy VLAN tag (***VLAN 30***). By default, this feature is disabled.

Feature Limitations:

- An overlap dynamic policy will fail when the overlap policy has filter (***F1***) and delivery interface (***D1***) on the same switch (***switch A***) and another delivery interface (***D2***) on another switch (***switch B***).
- Post-to-delivery dynamic policy will fail when it has a filter interface (**`F1`**) and a delivery interface (***D1***) on the same switch (***switch A***) and another delivery interface (***D2***) on another switch (***switch B***).
- Overlap policies may be reinstalled when a fabric port goes up or down when this feature is enabled.
- Double-tagged VLAN traffic is not supported and is dropped at the delivery interface.
- Tunnel interfaces are not supported with this feature.
- Only IPv4 traffic is supported; other non-IPv4 traffic is dropped at the delivery interface.
- Delivery interfaces with IP addresses (L3 delivery interfaces) are not supported.
- This feature is not supported on EOS switches (Arista 7280 switches).
- Delivery interface statistics may not be accurate when displayed using the **`sh policy`** command. This will happen when policy ***P1*** has ***F1***, ***D1***, ***D2*** and policy ***P2*** has ***F1***, ***D2***. In this case, overlap policy ***P1_o_P2*** is created with delivery interfaces ***D1***, ***D2***. Since ***D2*** is in both policies ***P1*** and ***P2***, overlap traffic is forwarded to ***D2*** with both the ***P1*** policy VLAN and the ***P2*** policy VLAN. The **`sh policy`** ***policy_name*** command will not show this doubling of traffic on delivery interface ***D2***. Delivery interface statistics will show this extra traffic forwarded from the delivery interface.

To enable this feature, enter the following command:

```
controller-1(config)# retain-user-policy-vlan
This will enable retain-user-policy-vlan feature. Non-IP packets will be dropped at delivery.
 Enter
"yes" (or "y") to continue: yes
```

To see the current Retain Policy VLAN configuration, enter the following command:

```
controller-1> show fabric
~~~~~~~~~~~~~~~~~~~~~~~~~ Aggregate Network State ~~~~~~~~~~~~~~~~~~~~~~~~~
Number of switches : 14
Inport masking : True
Number of unmanaged services : 0
Number of switches with service interfaces : 0
Match mode : l3-l4-offset-match
Number of switches with delivery interfaces : 11
Filter efficiency : 1:1
Uptime : 4 days, 8 hours
Max overlap policies (0=disable) : 10
Auto Delivery Interface Strip VLAN : True
Number of core interfaces : 134
State : Enabled
Max delivery BW (bps) : 2.18Tbps
Health : unhealthy
Track hosts : True
Number of filter interfaces : 70
Number of policies : 101
Start time : 2022-02-28 16:18:01.807000 UTC
Number of delivery interfaces : 104
Retain User Policy Vlan : True
```

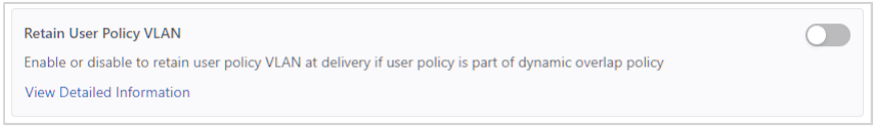
Use this feature with the **strip-second-vlan** option during delivery interface configuration to preserve the outer DMF fabric policy VLAN, strip the inner VLAN of traffic forwarded to a tool, or the **strip-no-vlan** option during delivery interface configuration.

12.11.2 Retain User Policy VLAN using the GUI

From the DMF Features page, proceed to the Retain User Policy VLAN feature card and perform the following steps to enable the feature.

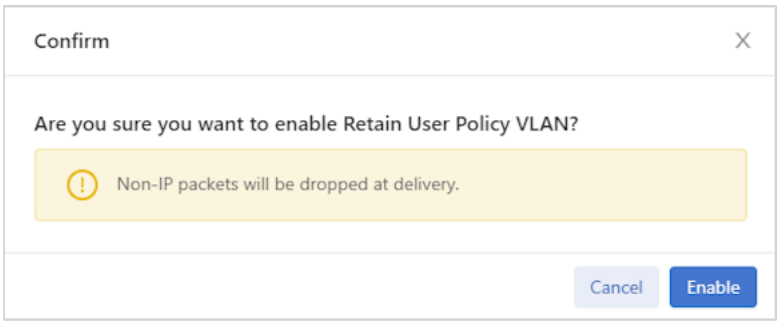
1. Select the **Retain User Policy VLAN** card.

   Figure 12-43: Retain User Policy VLAN Disabled



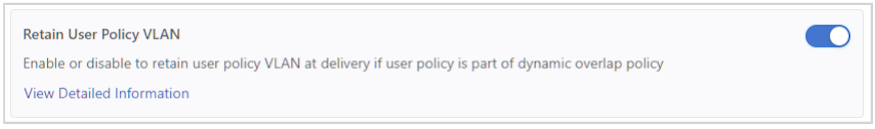
2. Toggle the Retain User Policy VLAN to **On**.
3. Confirm the activation by selecting **Enable**. Or, **Cancel** to return to the DMF Features page.

   Figure 12-44: Enable Retain User Policy VLAN



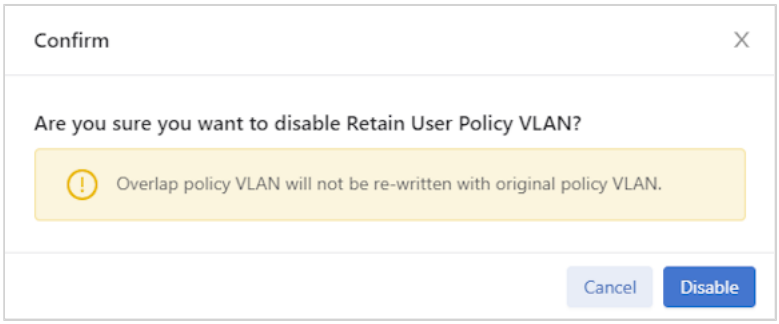
4. Retain User Policy VLAN is running.

   Figure 12-45: Retain User Policy VLAN Enabled



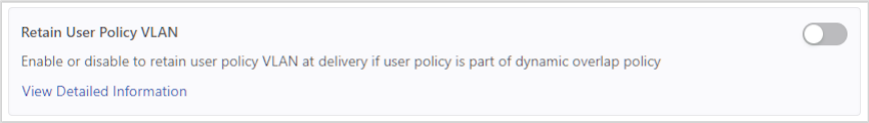
5. To disable the feature, toggle the Retain User Policy VLAN to **Off**. Select **Disable** and confirm.

   Figure 12-46: Disable Retain User Policy VLAN



The feature card updates with the status.

Figure 12-47: Retain User Policy VLAN Disabled



12.12 Tunneling

For more information about Tunneling please refer to the Understanding Tunneling section.

Enable and disable Tunneling using the steps described in the following topics.

- Configure Tunneling using the GUI
- Configure Tunneling using the CLI

12.12.1 Configuring Tunneling using the GUI

From the DMF Features page, proceed to the Tunneling feature card and perform the following steps to enable the feature.

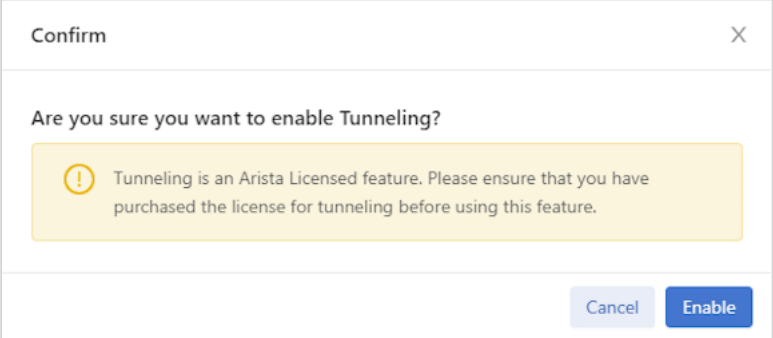
1. Select the **Tunneling** card.

Figure 12-48: Tunneling Disabled



2. Toggle Tunneling to **On**.
3. Confirm the activation by selecting **Enable**. Or, **Cancel** to return to the DMF Features page.

Figure 12-49: Enable Tunneling



Note: CRC Check must be running before attempting to enable Tunneling. An error message displays if CRC Check is not enabled. Proceeding to select Enable results in a validation error message. Refer to the CRC Check section for more information on configuring the CRC Check feature.

Figure 12-50: CRC Check Warning Message



4. Tunneling VLAN is running.

Figure 12-51: Tunneling Enabled



5. To disable the feature, toggle Tunneling to **Off**. Select **Disable** and confirm.

Figure 12-52: Disable Tunneling



The feature card updates with the status.

Figure 12-53: Tunneling VLAN Disabled



12.12.2 Configuring Tunneling using the CLI

To enable the Tunneling, enter the following command:

```
controller-1(config)# tunneling

Tunneling is an Arista Licensed feature.
Please ensure that you have purchased the license for tunneling before using this feature.
Enter "yes" (or "y") to continue: y
controller-1(config)#
```

To disable the Tunneling, enter the following command:

```
controller-1(config)# no tunneling
This would disable tunneling feature? Enter "yes" (or "y") to continue: y
controller-1(config)#
```

12.13 VLAN Preservation

In DANZ Monitoring Fabric (DMF), metadata is appended to the packets forwarded by the fabric to a tool attached to a delivery interface. This metadata is encoded primarily in the outer VLAN tag of the packets.

By default (using the auto-delivery-strip feature), this outer VLAN tag is always removed on egress upon delivery to a tool.

The VLAN preservation feature introduces a choice to selectively preserve a packet's outer VLAN tag instead of stripping or preserving all of it.

VLAN preservation works in both push-per-filter and push-per-policy mode for auto-assigned and user-configured VLANs.



Note: VLAN preservation applies to switches running SWL OS and does not apply to switches running EOS.

This functionality only supports 2000 VLAN IDs and port combinations per switch.

Support for VLAN preservation is on select Broadcom® switch ASICs. Ensure your switch model supports this feature before attempting to configure it.

12.13.1 Configure VLAN Preservation

VLAN preservation can be configured at global and local levels. A local configuration can override the global configuration. Follow the steps outlined below to configure a Global Configuration (steps 1 - 4), Local Configuration (steps 5-7), or an MLAG Delivery Interface configuration within an MLAG domain (step 8).

Global Configuration

1. To view or edit the global configuration, navigate to the DANZ Monitoring Fabric (DMF) Features page by selecting the **gear** icon in the navigation bar.

   Figure 12-54: DMF Menu Bar

   The DMF Feature allows for managing fabric-wide settings for DMF.

2. Scroll to the **VLAN Preservation** card.

   Figure 12-55: DMF Features Page

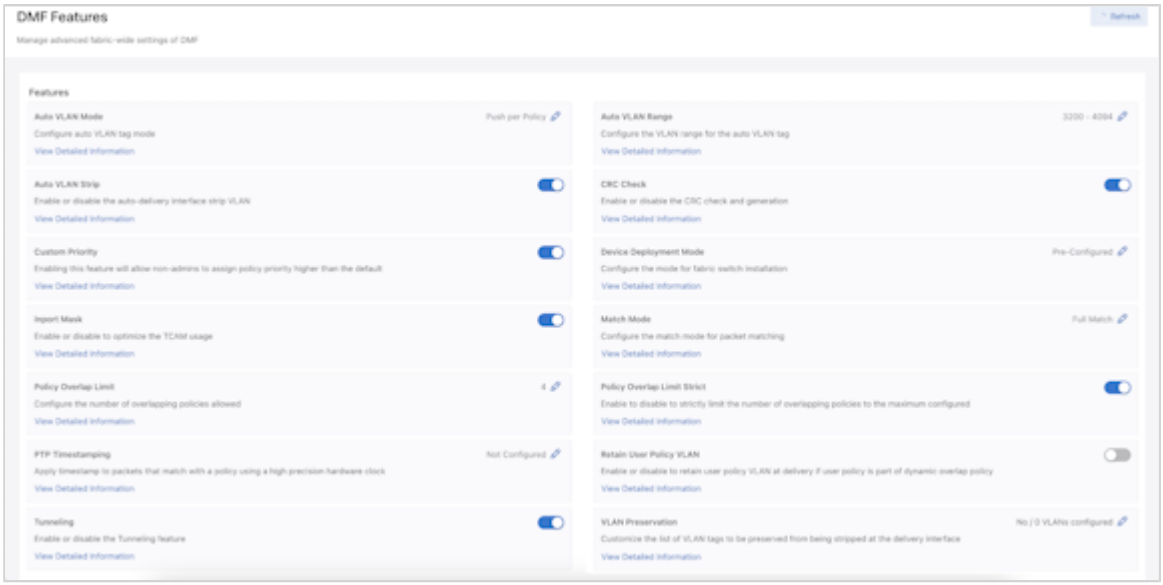
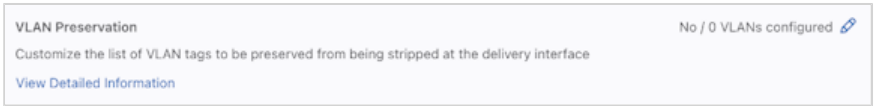
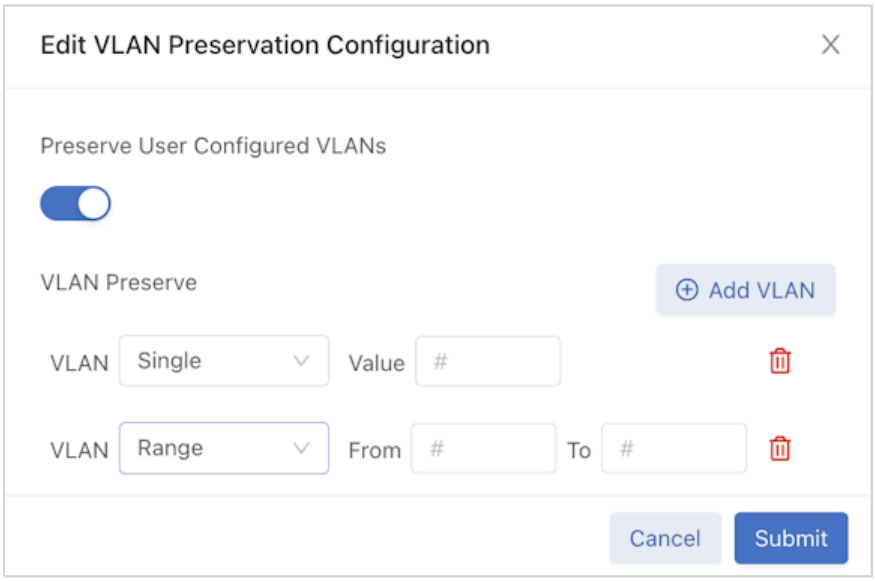
   

   Figure 12-56: VLAN Preservation Card

3. Select **Edit** (**pencil** icon) to configure or modify the global VLAN Preservation feature settings.

Figure 12-57: Edit VLAN Preservation Configuration



The edit screen has two input sections:

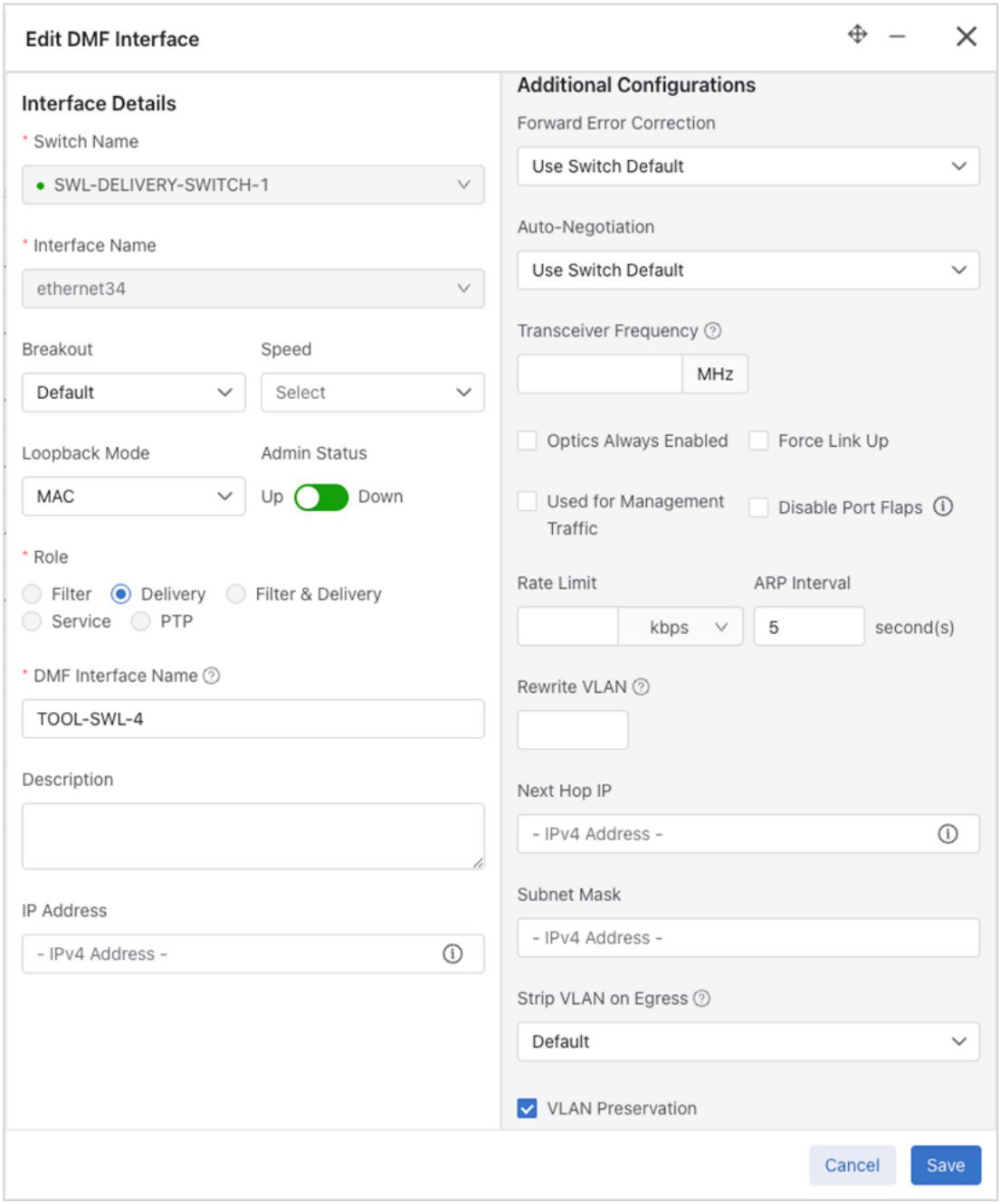
- Toggle **on** or **off** the **Preserve User Configured VLANs**.
- Enter the parameters for **VLAN Preserve** using the following functions:
  - Use **+ Add VLAN** to add VLAN IDs.
  - Select the **Single** VLAN type drop-down to add a single VLAN ID.
  - Select the **Range** VLAN type drop-down to add a continuous VLAN ID range.
  - Use the **Trash** icon (delete) to delete a single VLAN ID or a VLAN ID range.

4. Select **Submit** to save the configuration.

Local Configuration

5. The VLAN Preservation configuration can be applied per-delivery interface while configuring or editing a delivery or filter-and-delivery interface in **DMF Interfaces** and **Monitoring Interfaces** > **Delivery Interfaces**.

Figure 12-58: Monitoring Interfaces Delivery Interface Create Interface



6. The following inputs are available for the local feature configuration:
   - Enable **VLAN Preservation**. Use this option to preserve all user-configured VLAN IDs in push-per-policy or push-per-filter mode on a selected delivery interface. The packets with the user-configured VLANs will have their fabric-applied VLAN tags preserved even after leaving the respective delivery interface.
   - **Preserve User Configured VLANs**. Refer to Step **3**.
   - Disable **VLAN Preservation**. Disabling this option will ignore this feature configuration given globally/locally for this delivery interface. VLAN Preservation is enabled by default.
7. Select **Save** to save the configuration.

VLAN Preservation for MLAG Delivery Interfaces

8. Configure VLAN preservation for MLAG delivery interfaces using the **FabricMLAGs** page while configuring an MLAG Domain toggling the **VLAN Preservation** and **Preserve User Configured VLANs** switches to **on** (as required).

Figure 12-59: Create MLAG Domain

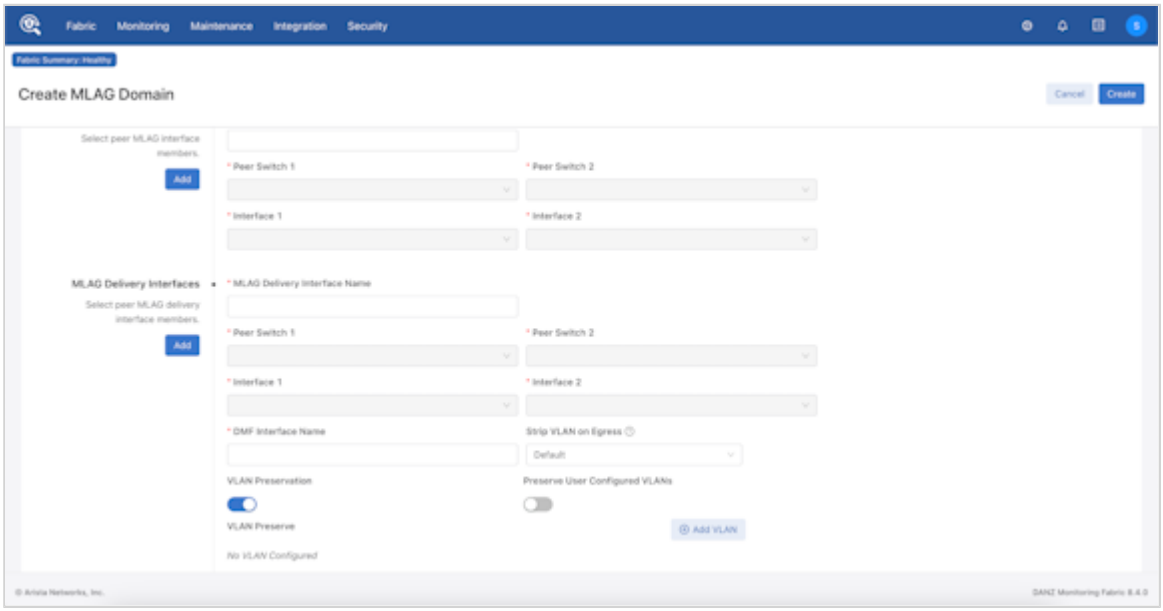


Figure 12-60: MLAG VLAN Preservation & Preserve User Configured VLANs (expanded view)



12.13.2 Using the CLI to Configure VLAN Preservation

Configure VLAN preservation at two levels: global and local. A local configuration can override the global configuration.

Global Configuration

Enable VLAN preservation globally using the vlan-preservation command from the config submode to apply a global configuration.

```
(config)# vlan-preservation
```

Two options exist while in the **config-vlan-preservation** submode:

- preserve-user-configured-vlans
- preserve-vlans

Use the help function to list the options by entering a **?** (question mark).

```
(config-vlan-preservation)# ?
Commands:
preserve-user-configured-vlans     Preserve all user-configured VLANs for all delivery interfaces
preserve-vlan                      Configure VLAN ID to preserve for all delivery interfaces
```

Use the **preserve-user-configured-vlans** option to preserve all user-configured VLANs. The packets with the user-configured VLANs will have their fabric-applied VLAN tags preserved even after leaving the respective delivery interface.

```
(config-vlan-preservation)# preserve-user-configured-vlans
```

Use the **preserve-vlan** option to specify and preserve a particular VLAN ID. Any VLAN ID may be provided. In the following example, the packets with VLAN ID 100 or 200 will have their fabric-applied VLAN tags preserved upon delivery to the tool.

```
(config-vlan-preservation)# preserve-vlan 100
(config-vlan-preservation)# preserve-vlan 200
```

Local Configuration

This feature applies to delivery and both-filter-and-delivery interface roles.

Fabric-applied VLAN tag preservation can be enabled locally on each delivery interface as an alternative to the global VLAN preservation configuration. To enable this functionality locally, enter the following configuration submode using the **if-vlan-preservation** command to specify either one of the two available options. Use the help function to list the options by entering a **?** (question mark).

```
(config-switch-if)# if-vlan-preservation
(config-switch-if-vlan-preservation)# ?
Commands:
preserve-user-configured-vlans     Preserve all user-configured VLANs for all delivery interfaces
preserve-vlan                      Configure VLAN ID to preserve for all delivery interfaces
```

Use the **preserve-user-configured-vlans** option to preserve all user-configured VLAN IDs in **push-per-policy** or **push-per-filter** mode on a selected delivery interface. All packets egressing such delivery interface will have their user-configured fabric VLAN tags preserved.

```
(config-switch-if-vlan-preservation)# preserve-user-configured-vlans
```

Use the **preserve-vlan** option to specify and preserve a particular VLAN ID. For example, if any packets with VLAN ID 100 or 300 egress the selected delivery interface, VLAN IDs 100 and 300 will be preserved.

```
(config-switch-if-vlan-preservation)# preserve-vlan 100
(config-switch-if-vlan-preservation)# preserve-vlan 300
```



Note: Any local vlan-preservation configuration overrides the global configuration for the selected interfaces by default.

On an MLAG delivery interface, the local configuration follows the same model, as shown below.

```
(config-mlag-domain-if)# if-vlan-preservation member role
(config-mlag-domain-if)# if-vlan-preservation
(config-mlag-domain-if-vlan-preservation)# preserve-user-configured-vlans preserve-vlan
```

To disable selective VLAN preservation for a particular delivery or both-filter-and-delivery interface, use the following command to disable the feature's global and local configuration for the selected interface:

```
(config-switch-if)# role delivery interface-name del
<cr>                 no-analytics          strip-no-vlan         strip-second-vlan
ip-address            no-vlan-preservation  strip-one-vlan        strip-two-vlan
(config-switch-if)# role delivery interface-name del no-vlan-preservation
```

CLI Show Commands

The following show command displays the device name on which VLAN preservation is enabled and the information about which VLAN is preserved on specific selected ports. Use the data in this table primarily for debugging purposes.

```
# show switch all table vlan-preserve
# Vlan-preserve Device name Entry key
-|-------------|-----------|---------------------|
1 0             delivery1   VlanVid(0x64), Port(6)
2 0             filter1     VlanVid(0x64), Port(6)
3 0             core1       VlanVid(0x64), Port(6)
```

12.13.3 Troubleshooting

Use the following commands to troubleshoot the scenario in which a tool attached to a delivery interface expects a packet with a preserved VLAN tag, but instead, there is no tag attached to it; double-check the following.

1. A partial policy installation may occur if any delivery interface fails to preserve the VLAN tag. This can happen when exceeding the 2000 VLAN ID/Port combination limit. Use the **`show policy`** ***policy-name*** command to obtain a detailed status, as shown in the following example:

```
(config)# show policy vlan-999
Policy Name                           : vlan-999
Config Status                         : active - forward
Runtime Status                        : installed but partial failure
Detailed Status                       : installed but partial failure -
                                        Failed to preserve VLAN's on some/all
                                        delivery interfaces, see warnings for details
Priority                              : 100
Overlap Priority                      : 0
# of switches with filter interfaces  : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces  : 0
# of filter interfaces                : 1
# of delivery interfaces              : 1
# of core interfaces                  : 2
# of services                         : 0
# of pre service interfaces           : 0
# of post service interfaces          : 0
Push VLAN                             : 999
Post Match Filter Traffic             : -
Total Delivery Rate                   : -
Total Pre Service Rate                : -
Total Post Service Rate               : -
Overlapping Policies                  : none
Component Policies                    : none
Installed Time                        : 2023-11-06 21:01:11 UTC
Installed Duration                    : 1 week
```

2. Verify the running config and review if the VLAN preservation configuration is enabled for that VLAN ID and on that delivery interface.

```
(config-vlan-preservation)# show running-config | grep "preserve"
! vlan-preservation
vlan-preservation
preserve-vlan 100
```

3. Verify the **`show switch`** ***`switch-name`*** **`table vlan-preserve`** output. It displays the ports and VLAN ID combinations that are enabled.

```
(config-policy)# show switch core1 table vlan-preserve
# Vlan-preserve Device name Entry key
-|-------------|-----------|---------------------|
1 0             core1       VlanVid(0x64), Port(6)
```

4. The same configuration can be verified from a switch (e.g., core1) by using the command below:

```
root@core1:~# ofad-ctl gt vlan_preserve
VLAN PRESERVE TABLE:
--------------------
VLAN:  100  Port:    6  PortClass:    6
```

5. Verify if a switch has any associated preserve VLAN warnings among the fabric warnings:

```
(config-vlan-preservation)# show fabric warnings | grep "preserve
1 delivery1 (00:00:52:54:00:85:ca:51) Switch 00:00:52:54:00:85:ca:51
cannot preserve VLANs for some interfaces due to resource exhaustion.
```

6. The **`show fabric warnings feature-unsupported-on-device`** command provides information on whether VLAN preservation is configured on any unsupported devices:

```
(config-switch)# show fabric warnings feature-unsupported-on-device
# Name Warning
-|----|------------------------------------------------------------|
1 del1 VLAN preservation feature is not supported on EOS switch eos
```

In the event of any preserve VLAN fabric warnings, please contact the Arista Support Team for assistance.

12.14 Reuse of Policy VLANs

Policies can reuse VLANs for policies in different switch islands. A switch island is an isolated fabric managed by a single pair of controllers; there is no data plane connection between fabrics in different switch islands. For example, with a single Controller pair managing six switches (***switch1***, ***switch2***, ***switch3***, ***switch4***, ***switch5***, and *switch6* the option exists to create two fabrics with three switches each (***switch1***, ***switch2***, ***switch3*** in one switch island and ***switch4***, ***switch5***, and ***switch6*** in another switch island), as long as there is no data plane connection between switches in the different switch islands.

There is no command needed to enable this feature. If the above condition is met, creating policies in each switch island with the same policy VLAN tag is supported.

In the condition mentioned above, assign the same policy VLAN to two policies in different switch islands using the **`push-vlan`** ***vlan-tag*** command under policy configuration. For example, policy ***P1*** in ***switch island 1*** assigned push-vlan ***10***, and policy ***P2*** in ***switch island 2*** assigned the same vlan tag ***10*** using push-vlan ***10*** under policy configuration.

When a data plane link connects two switch islands, it becomes one switch island. In that case, two policies cannot use the same policy vlan tag, so one of the policies (***P1*** or ***P2***) will become inactive.

12.15 Rewriting the VLAN ID for a Filter Interface

When sharing a destination tool with multiple filter interfaces, use the VLAN identifier assigned by the rewrite VLAN option to identify the ingress filter interface for specific packets. To use the rewrite VLAN option, assign a unique VLAN identifier to each filter interface. Ensure this VLAN ID is outside of the auto-VLAN range.



Note: In push-per-policy mode, enabling the rewrite VLAN feature on filter interfaces is impossible. When doing so, a validation error is displayed. This feature is available only in the push-per-filter mode.

The following commands change the VLAN tag on packets received on the interface ethernet10 on f-switch1 to 100. The **role** command in this example also assigns the alias **TAP-PORT-1** to Ethernet interface ***10***.

```
controller-1(config)# switch f-switch1
controller-1(config-switch-if)# interface ethernet10
controller-1(config-switch-if)# role filter interface-name TAP-PORT-1 rewrite vlan 100
```

The rewrite VLAN option overwrites the original VLAN frame tag if it was already tagged, and this changes the CRC checksum so it no longer matches the modified packet. The switch CRC option, enabled by default, rewrites the CRC after the frame has been modified so that a CRC error does not occur.



Note: Starting with ***DMF Release 7.1.0***, simultaneously rewriting the VLAN ID and MAC address is supported and uses VLAN rewriting to isolate traffic while using MAC rewriting to forward traffic to specific VMs.

12.16 Reusing Filter Interface VLAN IDs

A DMF fabric comprises groups of switches, known as islands, connected over the data plane. There are no data plane connections between switches in different islands. When Push-Per-Filter forwarding is enabled, monitored traffic is forwarded within an island using the VLAN ID affiliated with a Filter Interface. These VLAN IDs are configurable. Previously, the only recommended configuration was for these VLAN IDs to be globally unique.

This feature adds official support for associating the same VLAN ID with multiple Filter Interfaces as long as they are in different islands. This feature provides more flexibility when duplicating Filter Interface configurations across islands and helps prevent using all available VLAN IDs.

Note that within each island, VLAN IDs must still be unique, which means that Filter Interfaces in the same group of switches cannot have the same ID. When trying to reuse the same VLAN ID within an island, DMF generates a fabric error, and only the first Filter Interface (as sorted alphanumerically by DMF name) remains in use.

Configuration

This feature requires no special configuration beyond the existing Filter Interface configuration workflow.

Troubleshooting

A fabric error occurs if the same VLAN ID is configured more than once in the same island. The error message includes the Filter Interface name, the switch name, and the VLAN ID that is not unique. When encountering this error, pick a different non-conflicting VLAN ID.

Filter Interface invalid VLAN errors can be displayed in the CLI using the following command:

```
>show fabric errors filter-interface-invalid-vlan
------------------------------------------------------------------------ Invalid Filter Interface VLAN(s) ------------------------------------------------------------------------
# DMF Name   IF Name   Switch                          Rewrite VLAN Details
-|----------|---------|-------------------------------|------------|-----------------------------------------------------------------------------------------------------|
1 filter1-f1 ethernet2 filter1 (00:00:[redacted]bc) 1           The configured rewrite VLAN 1 for filter interface filter1-f1 is not unique within its fabric.
```

The following is a vertical representation of the CLI output above for illustrative purposes only.

```
>show fabric errors filter-interface-invalid-vlan
~~ Invalid Filter Interface VLAN(s) ~~
#                1
DMF Name         filter1-f1
IF Name          ethernet2
Switch           filter1 (00:00:52:54:00:4b:c9:bc)
Rewrite VLAN     1
Details          The configured rewrite VLAN 1 for filter interface filter1-f1
                 is not unique within its fabric.
```

It is helpful to know all of the switches in an island. The following command lists all of the islands (referred to in this command as **switch clusters**) and their switch members:

```
>show debug switch-cluster
# Member
-|--------------|
1 core1, filter1
```

It can also be helpful to know how the switches within an island are interconnected. Use the following command to display all the links between the switches:

```
>show link all
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Links ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Active State Src switch Src IF Name Dst switch Dst IF Name Link Type Since
-|------------|----------|-----------|----------|-----------|---------|-----------------------|
1 active       filter1    ethernet1   core1      ethernet1   normal    2023-05-24 22:31:39 UTC
2 active       core1      ethernet1   filter1    ethernet1   normal    2023-05-24 22:31:40 UTC
```

Considerations

- VLAN IDs must be unique within an island. Filter Interfaces in the same island with the same VLAN ID are not supported.
- This feature only applies to manually configured Filter Interface VLAN IDs. VLAN IDs that are automatically assigned are still unique across the entire fabric.

12.17 Using Push-per-filter Mode

The push-per-filter mode setting does not enable tag-based forwarding. Each filter interface is automatically assigned a VLAN ID; the default range is ***1*** to ***4094***. To change the range, use the **`auto-vlan-range`** command.

The option exists to manually assign a VLAN not included in the defined range to a filter interface.

To manually assign a VLAN to a filter interface in ***push-per-filter*** mode, complete the following steps:

1. Change the auto-vlan-range from the default (***1-4094***) to a limited range, as in the following example:

```
controller-1(config)# auto-vlan-range vlan-min 1 vlan-max 1000
```

   The example above configures the auto-VLAN feature to use VLAN IDs from ***1*** to ***1000***.
2. Assign a VLAN ID to the filter interface that is not in the range assigned to the auto-VLAN feature.

```
controller-1(config)# role filter interface-name TAP-1 rewrite vlan 1001
```

12.18 Tag-based Forwarding

The DANZ Monitoring Fabric (DMF) Controller configures each switch with forwarding paths based on the most efficient links between the incoming filter interface and the delivery interface, which is connected to analysis tools. The TCAM capacity of the fabric switches may limit the number of policies to configure. The Controller can also use VLAN tag-based forwarding, which reduces the TCAM resources required to implement a policy.

Tag-based forwarding is automatically enabled when the auto-VLAN Mode is push-per-policy, which is the default. This configuration improves traffic forwarding within the monitoring fabric. DMF uses the assigned VLAN tags to forward traffic to the correct delivery interface, saving TCAM space. This feature is handy when using switches based on the Tomahawk chipset because these switches have higher throughput but reduced TCAM space.

12.19 Policy Rule Optimization

12.19.1 Prefix Optimization

A policy can match with a large number of IPv4 or IPv6 addresses. These matches can be configured explicitly on each match rule, or the match rules can use an address group. With prefix optimization based on IPv4, IPv6, and TCP ports, DANZ Monitoring Fabric (DMF) uses efficient masking algorithms to minimize the number of flow entries in hardware.

Example 1: Optimize the same mask addresses.

```
controller-1(config)# policy ip-addr-optimization
                controller-1(config-policy)# action forward
                controller-1(config-policy)# delivery-interface TOOL-PORT-1
                controller-1(config-policy)# filter-interface TAP-PORT-1
                controller-1(config-policy)# 10 match ip dst-ip 1.1.1.0 255.255.255.255
                controller-1(config-policy)# 11 match ip dst-ip 1.1.1.1 255.255.255.255
                controller-1(config-policy)# 12 match ip dst-ip 1.1.1.2 255.255.255.255
                controller-1(config-policy)# 13 match ip dst-ip 1.1.1.3 255.255.255.255
                controller-1(config-policy)# show policy ip-addr-optimization optimized-match
                Optimized Matches :
                10 ether-type 2048 dst-ip 1.1.1.0 255.255.255.252
```

Example 2: In this case, if a generic prefix exists, all the specific addresses are not programmed in TCAM.

```
controller-1(config)# policy ip-addr-optimization
                controller-1(config-policy)# action forward
                controller-1(config-policy)# delivery-interface TOOL-PORT-1
                controller-1(config-policy)# filter-interface TAP-PORT-1
                controller-1(config-policy)# 10 match ip dst-ip 1.1.1.0 255.255.255.255
                controller-1(config-policy)# 11 match ip dst-ip 1.1.1.1 255.255.255.255
                controller-1(config-policy)# 12 match ip dst-ip 1.1.1.2 255.255.255.255
                controller-1(config-policy)# 13 match ip dst-ip 1.1.1.3 255.255.255.255
                controller-1(config-policy)# 100 match ip dst-ip 1.1.0.0 255.255.0.0
                controller-1(config-policy)# show policy ip-addr-optimization optimized-match
                Optimized Matches :
                100 ether-type 2048 dst-ip 1.1.0.0 255.255.0.0
```

Example 3: IPv6 prefix optimization. In this case, if a generic prefix exists, the specific addresses are not programmed in the TCAM.

```
controller-1(config)# policy ip-addr-optimization
                controller-1(config-policy)# 25 match ip6 src-ip 2001::100:100:100:0 FFFF:FFFF:FFF
F::0:0
                controller-1(config-policy)# 30 match ip6 src-ip 2001::100:100:100:0 FFFF:FFFF::0
                controller-1(config-policy)# show policy ip-addr-optimization optimized-match
                Optimized Matches :
                30 ether-type 34525 src-ip 2001::100:100:100:0 FFFF:FFFF::0
```

Example 4: Different subnet prefix optimization. In this case, addresses belonging to different subnets are optimized.

```
controller-1(config)# policy ip-addr-optimization
                controller-1(config-policy)# 10 match ip dst-ip 2.1.0.0 255.255.0.0
                controller-1(config-policy)# 11 match ip dst-ip 3.1.0.0 255.255.0.0
                controller-1(config-policy)# show policy ip-addr-optimization optimized-match
                Optimized Matches : 10 ether-type 2048 dst-ip 2.1.0.0 254.255.0.0
```

12.19.2 Transport Port Range and VLAN Range Optimization

The DANZ Monitoring Fabric (DMF) optimizes transport port ranges and VLAN ranges within a single match rule. Improvements in DMF now support cross-match rule optimization.

Show Commands

To view the optimized match rule, use the show command:

```
# show policy policy-name optimized-match
```

To view the configured match rules, use the following command:

```
# show running-config policy policy-name
```

Consider the following DMF policy configuration.

```
# show running-config policy p1
! policy
policy p1
action forward
delivery-interface d1
filter-interface f1
1 match ip vlan-id-range 1 4
2 match ip vlan-id-range 5 8
3 match ip vlan-id-range 7 16
4 match ip vlan-id-range 10 12
```

With the above policy configuration and before the DMF 8.5.0 release, the four match conditions would be optimized into the following TCAM rules:

```
# show policy p1 optimized-match
Optimized Matches :
1 ether-type 2048 vlan 0 vlan-mask 4092
1 ether-type 2048 vlan 4 vlan-mask 4095
2 ether-type 2048 vlan 5 vlan-mask 4095
2 ether-type 2048 vlan 6 vlan-mask 4094
3 ether-type 2048 vlan 16 vlan-mask 4095
3 ether-type 2048 vlan 8 vlan-mask 4088
```

However, with the cross-match rule optimizations introduced in this release, the rules installed in the switch would further optimize TCAM usage, resulting in:

```
# show policy p1 optimized-match
Optimized Matches :
1 ether-type 2048 vlan 0 vlan-mask 4080
1 ether-type 2048 vlan 16 vlan-mask 4095
```

A similar optimization technique applies to L4 ports in match conditions:

```
# show running-config policy p1
! policy
policy p1
action forward
delivery-interface d1
filter-interface f1
1 match tcp range-src-port 1 4
2 match tcp range-src-port 5 8
3 match tcp range-src-port 7 16
4 match tcp range-src-port 9 14

# show policy p1 optimized-match
Optimized Matches :
1 ether-type 2048 ip-proto 6 src-port 0 -16
1 ether-type 2048 ip-proto 6 src-port 16 -1
```

12.20 Switch Dual Management Port

12.20.1 Overview

When a DANZ Monitoring Fabric (DMF) switch disconnects from the Controller, the switch is taken out of the fabric, causing service interruptions. The dual management feature solves the problem by providing physical redundancy of the switch-to-controller management connection. DMF achieves this by allocating a switch data path port to be bonded with its existing management interface, thereby acting as a standby management interface. Hence, it eliminates a single-point failure in the management connectivity between the switch and the Controller.

Once an interface on a switch is configured for management, this configuration persists across reboots and upgrades until explicitly disabling the management configuration on the Controller.

Configure an interface for dual management using the CLI or the GUI.



Note: Along with the configuration on the Controller detailed below, dual management requires a physical connection in the same subnet as the primary management link from the data port to a management switch.

12.20.2 Configuring Dual Management Using the CLI

1. From config mode, specify the switch to be configured with dual management, as in the following example:

```
Controller-1(config)# switch DMF-SWITCH-1
Controller-1(config-switch)#
```

   The CLI changes to the config-switch submode, to configure the specified switch.

2. From ***config-switch*** mode, enter the **`interface`** command to specify the interface to be configured as the standby management interface:

```
Controller-1(config-switch)# interface ethernet40
Controller-1(config-switch-if)#
```

   The CLI changes to the config-switch-if submode, to configure the specified interface.

3. From ***config-switch-if*** mode, enter the **`management`** command to specify the role for the interface:

```
Controller-1(config-switch-if)# management
Controller-1(config-switch-if)#
```

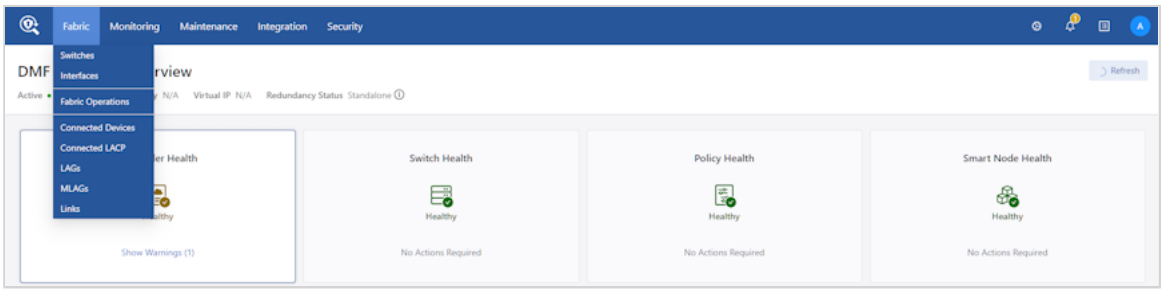


Note: When assigning an interface to a management role, no other interface-specific commands are honored for that interface (e.g., **`shut-down`**, **`role`**, **`speed`**, etc.).

12.20.3 Configuring Dual Management Using the GUI

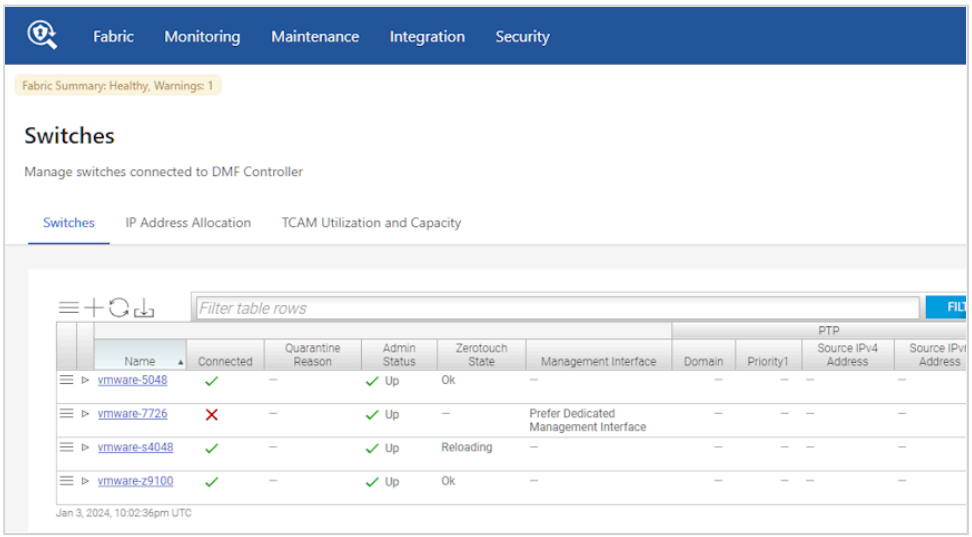
1. Select **Fabric > Switches** from the main menu.

Figure 12-61: Controller GUI Showing Fabric Menu List



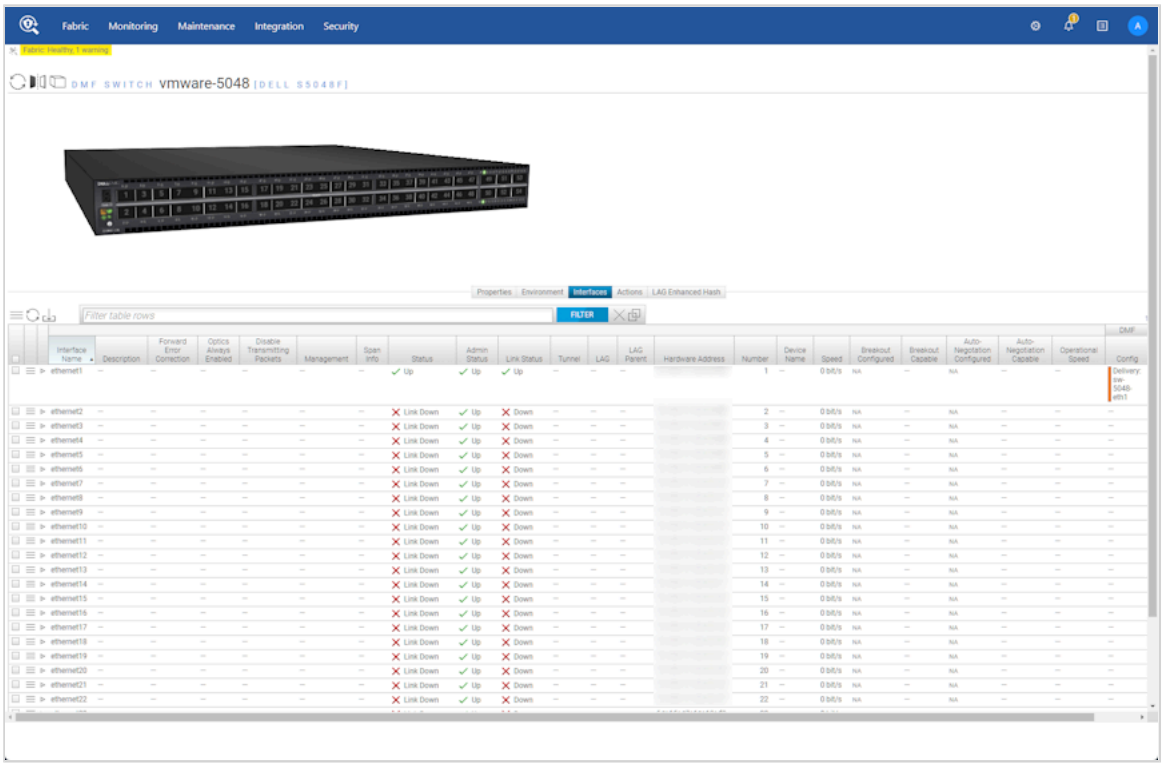
2. Select the switch name to be configured with dual management.

Figure 12-62: Controller GUI Showing Inventory of Switches



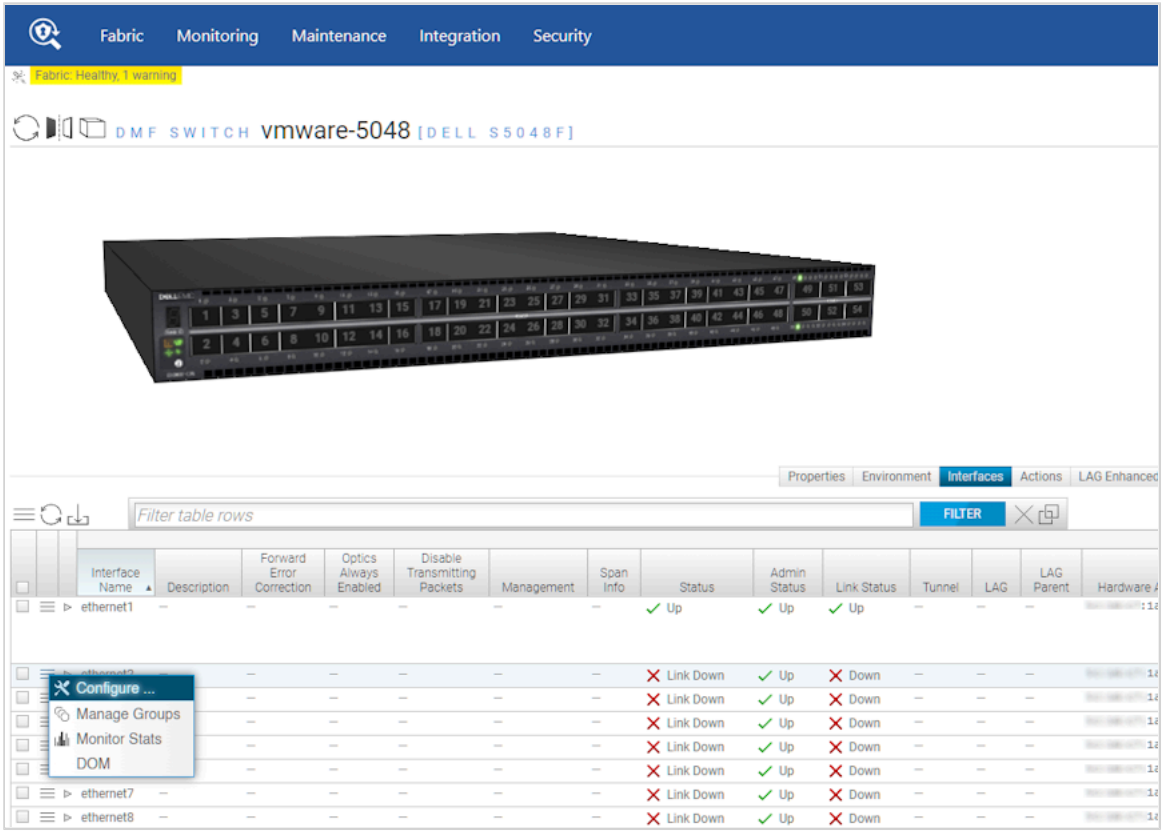
3. Select the **Interfaces** tab.

Figure 12-63: Controller GUI Showing Switch Interfaces



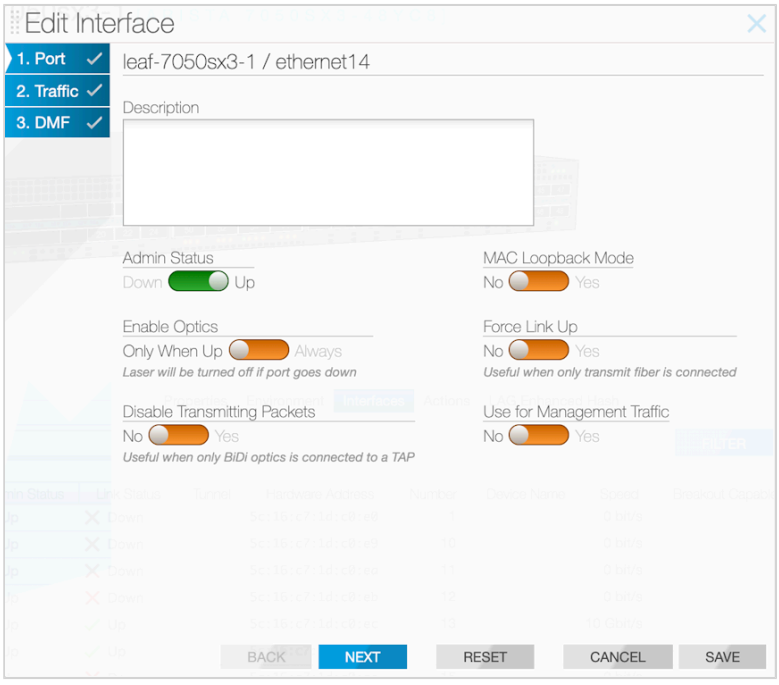
4. Identify the interface to be configured as the standby management interface.

Figure 12-64: Controller GUI Showing Configure Knob



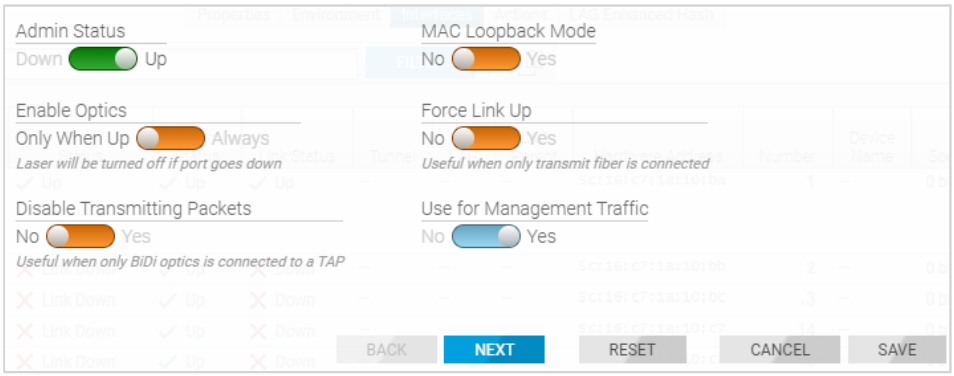
5. Select **Menu** to the left of the identified interface, then select **Configure**.

Figure 12-65: Controller GUI Showing Interface Settings



6. Set **Use for Management Traffic** to **Yes**. This action configures the interface to the standby management role.

Figure 12-66: Use for Management Traffic



7. Select **Save**.

12.20.4 Management Interface Selection Using the GUI

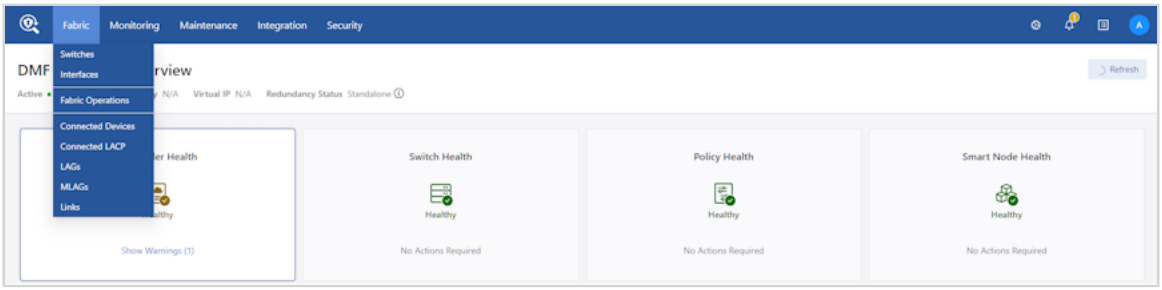
By default, the dedicated management interface serves as the management port, with the front panel data port acting as a backup only when the management interface is unavailable:

- When the dedicated management interface fails, the front panel data port becomes active as the management port.
- When the dedicated management interface returns, it becomes the active management port.
- When the management network is undependable, this can lead to switch disconnects.

The **Management Interface** choice dictates what happens when the management interface returns after a failover. Make this selection using the GUI or the CLI.

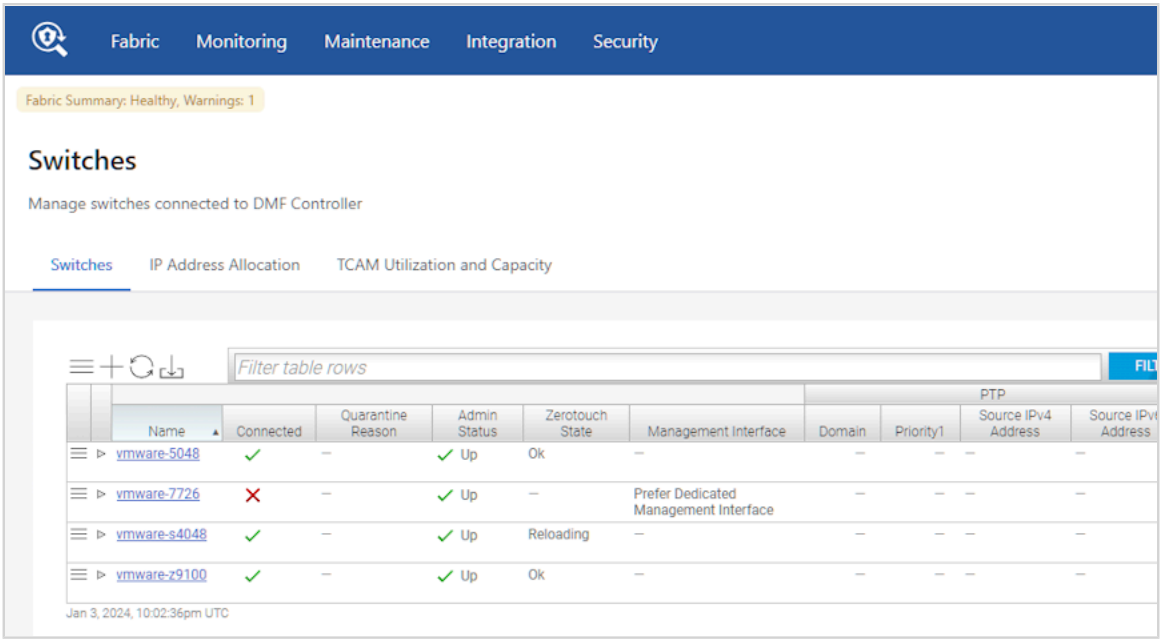
Select **Fabric > Switches**.

Figure 12-67: Fabric Switches



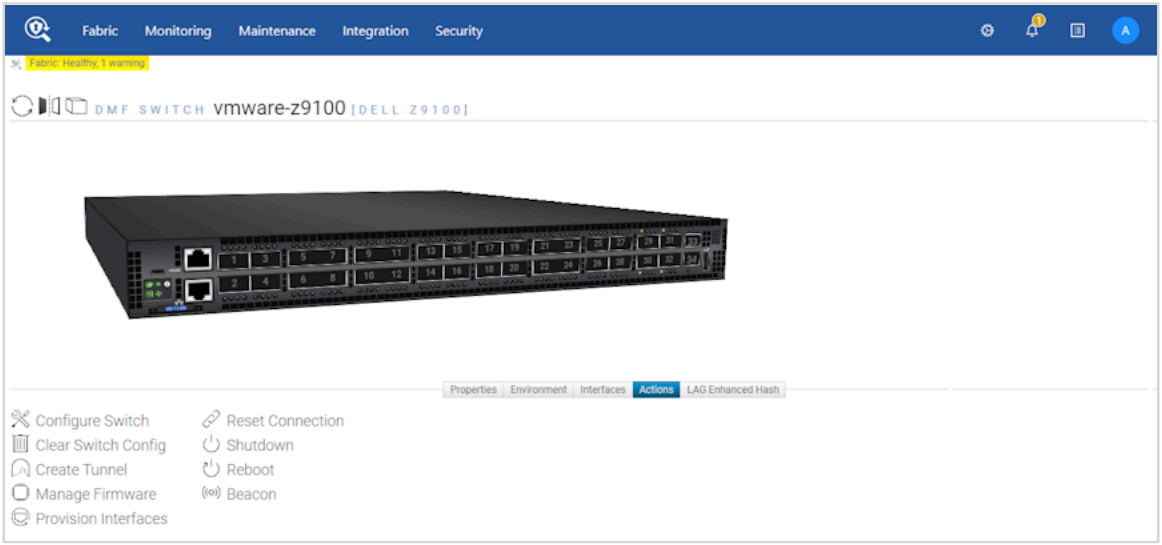
Select the switch name to be configured.

Figure 12-68: Switch Inventory



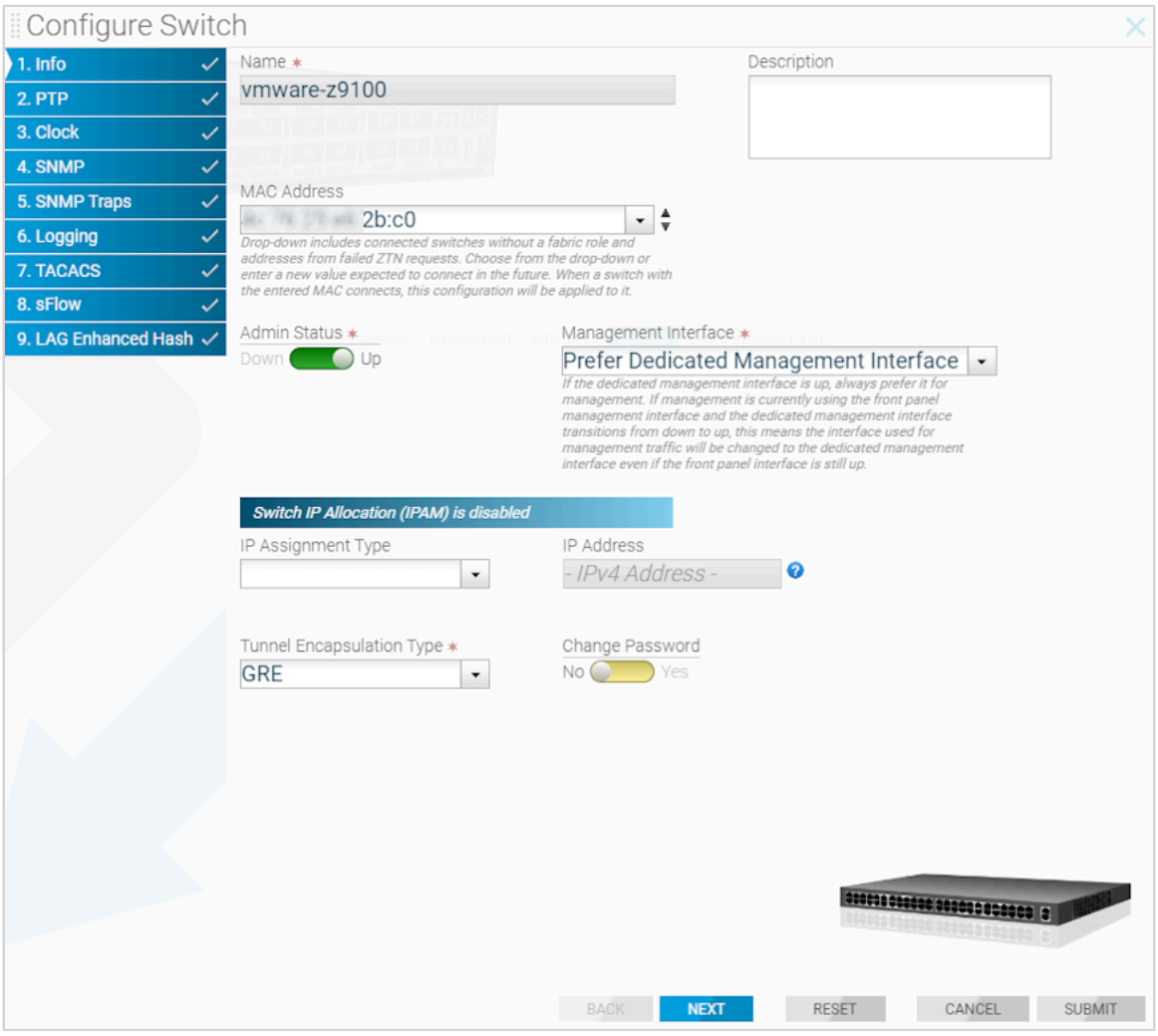
Select the **Actions** tab.

Figure 12-69: Switch Actions



Select **Configure Switch** and choose the required **Management Interface** setting.

Figure 12-70: Controller GUI Showing Dual Management Settings



If you select **Prefer Dedicated Management Interface** (the default), when the dedicated management interface goes down, the front panel data port becomes the active management port for the switch. When the dedicated management port comes back up, the dedicated management port becomes the active management port again, putting the front panel data port in an admin down state.

If you select **Prefer Current Interface**, when the dedicated management interface goes down, the front panel data port still becomes the active management port for the switch. However, when the dedicated management port comes back up, the front panel data port continues to be the active management port.

12.20.5 Management Interface Selection Using the CLI

By default, the dedicated management interface serves as the management port, with the front panel data port acting as a backup only when the management interface is unavailable:

- When the dedicated management interface fails, the front panel data port becomes active as the management port.
- When the dedicated management interface returns, it becomes the active management port.

When the management network is undependable, this can lead to switch disconnects. The **`management interface`** selection choice dictates what happens when the management interface returns after a failover.

```
Controller-1(config)# switch DMF-SWITCH-1
Controller-1(config-switch)#management-interface-selection ?
prefer-current-interface Set management interface selection algorithm
prefer-dedicated-management-interface Set management interface selection algorithm (default
 selection)
Controller-1(config-switch)#
```

If you select **prefer-dedicated-management-interface** (the default), when the dedicated management interface goes down, the front panel data port becomes the active management port for the switch. When the dedicated management port comes back up, the dedicated management port becomes the active management port again, putting the front panel data port in an admin down state.

If you select **prefer-current-interface**, when the dedicated management interface goes down, the front panel data port still becomes the active management port for the switch. However, when the dedicated management port comes back up, the front panel data port continues to be the active management port.

12.20.6 Switch Fabric Management Redundancy Status

To check the status of all switches configured with dual management as well as the interface that is being actively used for management, enter the following command in the CLI:

```
Controller-1# show switch all mgmt-stats
```

12.20.7 Additional Notes

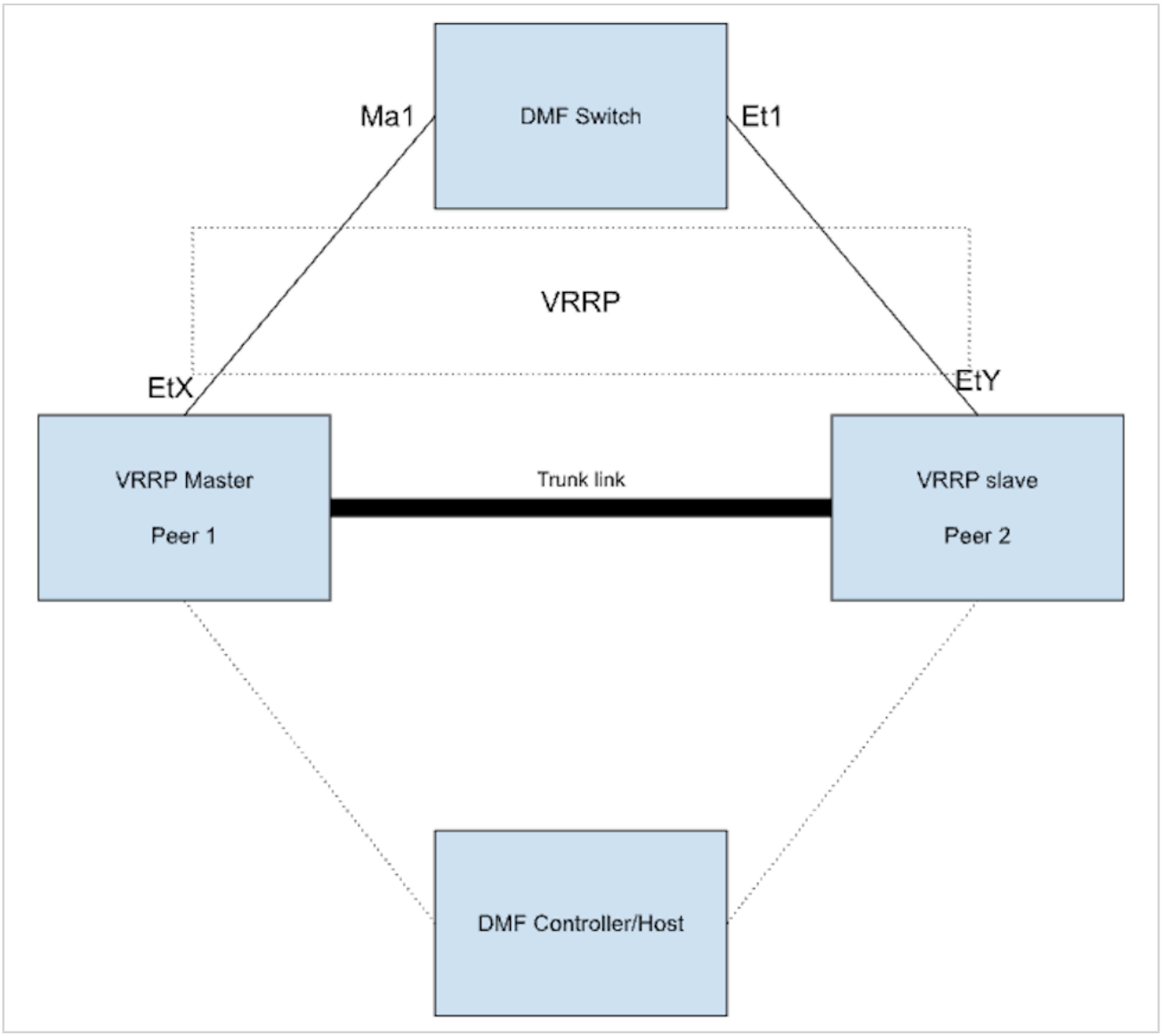
- A maximum of one data-plane interface on a switch can be configured as a standby management interface.
- The switch management interface **ma1** is a bond interface, having **oma1** as the primary link and the data plane interface as the secondary link.
- The bandwidth of the data-plane interface is limited regardless of the physical speed of the interface. Arista Networks recommends immediate remediation when the **oma1** link fails.

12.20.8 Management Redundancy on EOS Fixed System Chassis

Danz Monitoring Fabric (DMF) 8.7.0 provides support for Management Redundancy on an Extensible Operating System (EOS) Fixed System Chassis. It provides a method to enable redundant active/active connectivity on the management IP address for a DMF switch in a fixed system chassis using an out-of-band management port and a front-panel port on the switch.

The feature utilizes first hop redundancy protocols such as Virtual Router Redundancy Protocol (VRRP) running in the gateway devices and addressless forwarding on the switch.

Figure 12-71: Example - Management Redundancy



After enabling the feature, a floating loopback interface is created on the DMF switch and assigned with the Ma1 (Management interface) IP address. Addressless forwarding is enabled on Ma1 and the redundant port (e.g., Et1) along with proxy ARP. This supports the feature without using any more IP addresses. A default route is programmed on the DMF switch to point to the gateway IP (which is assumed to be the virtual IP address of the first-hop redundancy protocol).

When the feature is disabled, the system deletes the floating loopback interface on the DMF switch, and the original Ma1 configurations are automatically re-configured for the Ma1 interface.

Management Redundancy on an EOS fixed system chassis is supported on the following platforms:

- DCS-7280R/R2/R3
- DCS-7050PX4-32S
- DCS-7050DX4-32S

Configure the feature using one of the following:

- Configuring Management Redundancy using the CLI
- Configuring Management Redundancy using the GUI

12.20.8.1 Configuring Management Redundancy using the CLI

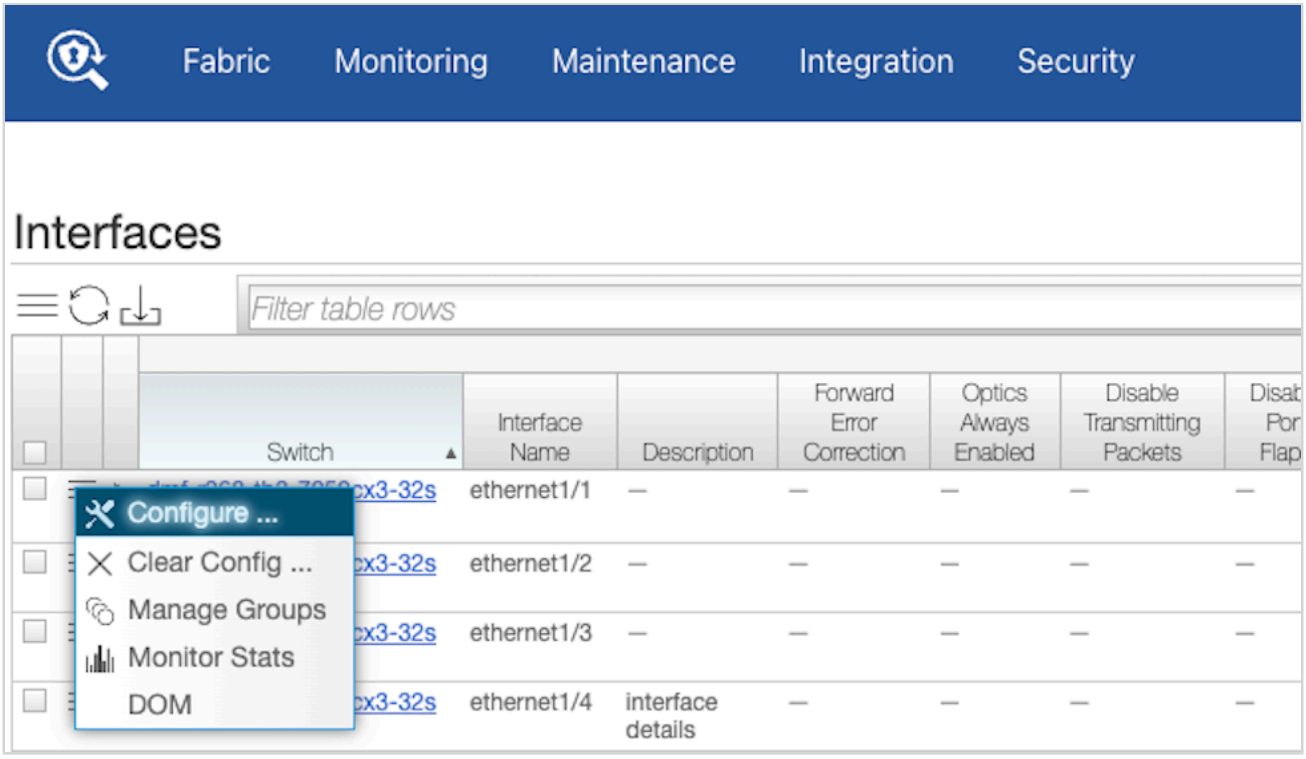
To configure the redundant management interface on the fixed system EOS switch, enter the following commands:

```
c1(config)# switch fixed-eos
c1(config-switch)# interface Ethernet1
c1(config-switch-if)# management
```

12.20.8.2 Configuring Management Redundancy using the GUI

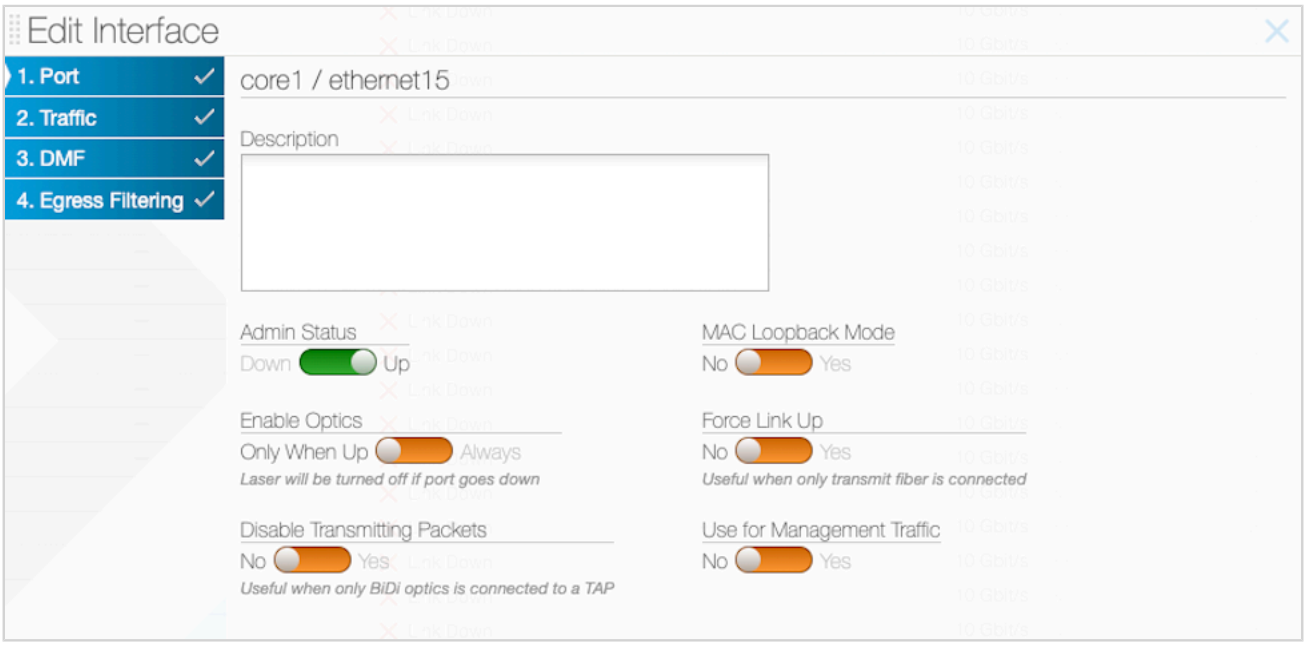
Navigate to **Fabric** > **Interfaces** and select **Configure** using the menu icon.

Figure 12-72: Configure Interface



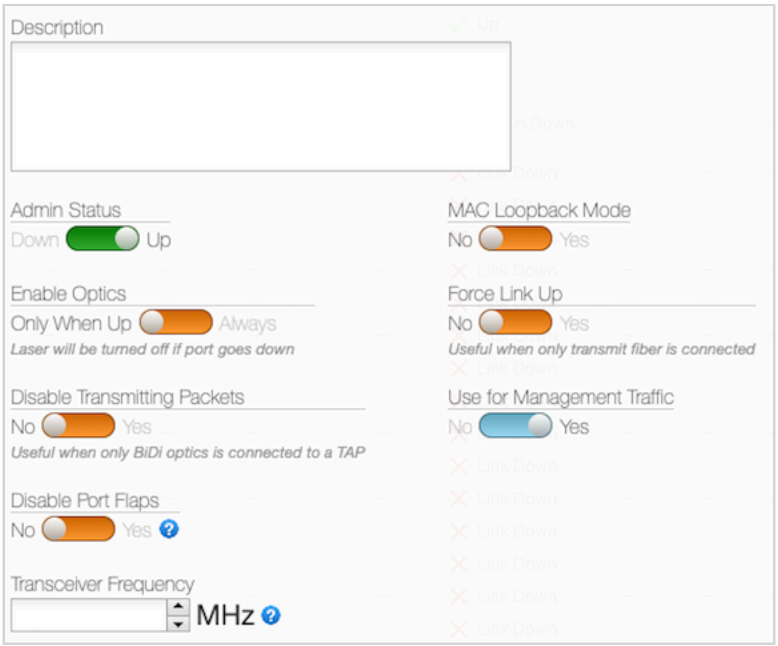
Navigate to **Edit Interface** > **Port**.

Figure 12-73: Edit Interface



Set **Use for Management Traffic** to **Yes**.

Figure 12-74: Use for Management Traffic



Select **Save**.

12.20.8.3 Show Commands

Several **show** commands are available to view the configuration settings of a redundant management interface.

1. Use the **show running-config switch *switch-name*** command to view the running-config and the management traffic interface settings, as shown in the following example:

```
c1(config)# show running-config switch fixed-eos

! switch
switch fixed-eos
  mac 28:99:3a:34:42:81
  !
  interface Ethernet1
    management
```

2. Use the **show switch *switch-name* running-config** command to view the **running-config** generated for the switch, as shown in the following example:

```
c1(config)# show switch fixed-eos running-config
!
management dmf
controller address 10.243.255.239
no disabled
hostname fixed-eos
username controller privilege 15 role network-admin secret sha512
$6$XTUb68lm$0ohxyre2dzkRoB9ycR37Tjy/sA/DR8V3fOpwdFSjcjy2FmB3GIyOX2T5q.JztN9/tz7ZW2VU
SCyPTg8I5dDtR1
aaa authorization exec default local
username admin ssh-key ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQC6Bll8rjahrn2YTQ9sQbuQXkt9+KMxJWIq3+d2M6+ZGLeQ5nEMdaA
LQ2pSoO2DqOBvHD0zq
7qBVDzOKAEkUxdVeFuSZVcuTss32NGKU+q2oD/5Gxu+4tOWIo7J6ljWEtF6FqzINRf7mlEFpSBShLBiX5CfDC
vHhwfI4fEwNGDsWS
UgN19j/uWwugv3nuPulCt+2xIUPhRhKHcck6qN+1LTJFJ1VQrljqYq65lDIJ3PxgN9ML1+imbnx4w4kgUXEWRNE
RkZvqS7nhV1U9d
vblq824eBaM3KnsgDxYDEHu3PAJZLjerqu426v/mC3J/+2ghbcfADwSww0/4rYaKS1Btt4SWG+qVDwtQSKPF9HGVz
1+qlhnmGJ+02
WfqXc+qNF5Sa9RpJIJ68VMjHGIHRhOGy6CpI8kA1de2RI7xz7lgB1/3M7DYd5mdV7jlR/Bga9jWwAR60hTlzUYJrBQF
J30DIjvyT3
ZNzgmgS3Bi/fytRRYz4DjkrBNMMggXnk0kIZALxnaabPK14esInW5HMJEapMVgLraOwgGlq+8M8BfCSXbvZvAEFSNKhAE
4hhoTjQ0
ds5AuVx8KZODZRnlwtzLRwIv12SCVEhT/hQufM4LDI+Aeek1WNzETAKRlZ/As21ni4+RaKGgFtfpkIp/dyviYTVinNm154S
LtHzjh
XeIp6rQ== floodlight@controller\n
clock timezone UTC
ntp server 10.243.255.239
ntp server ntp1.aristanetworks.com
ntp server ntp2.aristanetworks.com
ntp server ntp3.aristanetworks.com
ntp server ntp4.aristanetworks.com
logging host 10.243.255.239
logging host docker1.eng.bigswitch.com 11514
logging trap
snmp-server engineID local 800092a20328993a344281
! redundant management interface Ethernet1
```

In the example, the output shows **! redundant management interface Ethernet1** is destined for the switch. This isn't an actual configuration in the usual sense; it's a comment (i.e. **!** is at the front of the string). This line is parsed on the EOS switch and replaced with the relevant configuration to tune the redundant management interface.

3. If the line **! redundant management interface Ethernet1** is not present in the running-config of the switch, use the **show fabric warnings** command to see if there was an issue in applying the

configuration to that switch interface. The following example displays the output of the **show fabric warnings** command when the switch is a VM. DMF does not support this feature for EOS VMs.

```
c1(config)# show fabric warnings
# Name  Warning
-|-----|--------------------------------------------------------------------------------|
1 core2 Redundant management interface config not supported on VM based EOS systems
```

4. To confirm that management interface redundancy has been configured on EOS, use the **show switch all mgmt-stats** command. This will display the management redundancy operational-state and the interface names that are involved.

```
c1# show switch all mgmt-stats
~~~~~~~~~~~~~~ Switches ~~~~~~~~~~~~~~
# Switch DPID               Redundant
-|-------------------------|---------|
1 swl                       False
2 eos                       True


~~~~~~~~~~~~~~~~~~ Interfaces of Switch DPIDs  ~~~~~~~~~~~~~~~~~~
# Switch DPID               Interface   Active Fail Count LinkUp
-|-------------------------|-----------|------|----------|------|
1 swl                       ethernet2   False  1          False
2 swl                       oma1        True   0          True
3 eos                       Ethernet9   True   3          True
4 eos                       Management1 True   8          True
```

5. In the earlier example, the switch **eos** shows the operational state of management redundancy. The redundant interfaces are **Ethernet9** and **Management1**.

12.20.8.4 Troubleshooting

If the fixed-system EOS switch is not using a front-panel port as an alternative management redundancy port, yet it was configured on the Controller, perform the following steps:

1. Inspect the output of the **`show fabric warnings`** command to see if the config has been filtered.

```
c1(config)# show fabric warnings
# Name  Warning
-|-----|--------------------------------------------------------------------------------|
1 core2 Redundant management interface config not supported on VM based EOS systems
```

2. Inspect the output of the **`show switch all zerotouch`** command to confirm the switch is in an OK state. If it is not, this might be a pointer to the life-cycle managers on the switch having difficulty syncing up the updated running-config. In the following example, the switch named **`fixed-eos`** is in a **`reloading`** state. If this persists, perform step 3.

```
c1(config)# show switch all zerotouch
# Name      Device                          Ip address     Platform                   Serial number                     Zerotouch state Last update
-|---------|-------------------------------|--------------|--------------------------|---------------------------------|---------------|----------------
--------------|
1 fixed-eos 28:99:3a:34:42:81 (Arista)    172.30.157.207 x86_64-dcs-7020tr-48-eos   SSJ17251623                       reloading       2024-06-05
 16:02:38.041000 UTC
2 core2     52:54:00:b7:e7:ad (Linux KVM) 10.243.252.111 x86_64-veos-eos            2A7BE3544FA18468729F340F7A451506 ok              2024-06-05
 16:02:18.961000 UTC
3 core1     52:54:00:c5:54:58 (Linux KVM) 10.243.253.42  x86-64-bigswitch-bs3240-r0 525400c55458                      ok              2024-06-05
 16:02:26.504000 UTC
```

3. Investigate the **`show logging controller`** output to see if there's an issue with ZTN in trying to sync up the configuration.

```
c1(config)# show logging controller last hour

2024-06-13T10:13:07.699-07:00 floodlight: INFO
[PackedFileStateRepository:packed-file-state-repository-worker-thread] PFSTATEREP1007:
Beginning scan of segment files; type=integrity check;
enqueue-time=2024-06-13T17:13:07.699Z, dequeue-time=2024-06-13T10:13:07.699-07:00

2024-06-13T10:13:07.702-07:00 floodlight: INFO
[PackedFileStateRepository:packed-file-state-repository-worker-thread] PFSTATEREP1008:
Scanning segment file: /var/lib/floodlight/db/global-config/data/20240613021314.seg

2024-06-13T10:13:07.704-07:00 floodlight: INFO
[PackedFileStateRepository:packed-file-state-repository-worker-thread] PFSTATEREP1009:
Finished scan of segment files; type=integrity check; enqueue-time=2024-06-13T17:13:07.699Z,
dequeue-time=2024-06-13T17:13:07.699Z, complete-time=2024-06-13T10:13:07.704-07:00

2024-06-13T10:17:24.667-07:00 floodlight: INFO
[AbstractServiceAddressDirectory:FLTP-1-10] SERVADDR1216: Starting garbage collection

2024-06-13T10:17:24.671-07:00 floodlight: INFO
[AbstractServiceAddressDirectory:FLTP-1-10] SERVADDR1215: Scheduling garbage
collection: delay=31 minutes, reason=ROUTINE

2024-06-13T10:43:07.700-07:00 floodlight: INFO
[PackedFileStateRepository:packed-file-state-repository-worker-thread] PFSTATEREP1007:
Beginning scan of segment files; type=integrity check;
enqueue-time=2024-06-13T17:43:07.700Z, dequeue-time=2024-06-13T10:43:07.700-07:00

2024-06-13T10:43:07.703-07:00 floodlight: INFO
[PackedFileStateRepository:packed-file-state-repository-worker-thread] PFSTATEREP1008:
Scanning segment file: /var/lib/floodlight/db/global-config/data/20240613021314.seg

2024-06-13T10:43:07.705-07:00 floodlight: INFO
[PackedFileStateRepository:packed-file-state-repository-worker-thread] PFSTATEREP1009:
Finished scan of segment files; type=integrity check;
enqueue-time=2024-06-13T17:43:07.700Z, dequeue-time=2024-06-13T17:43:07.700Z,
complete-time=2024-06-13T10:43:07.705-07:00

2024-06-13T10:48:24.671-07:00 floodlight: INFO
[AbstractServiceAddressDirectory:FLTP-1-7] SERVADDR1216: Starting garbage collection

2024-06-13T10:48:24.677-07:00 floodlight: INFO
[AbstractServiceAddressDirectory:FLTP-1-7] SERVADDR1215: Scheduling garbage
collection: delay=31 minutes, reason=ROUTINE
```

Arista Technical Support can investigate the floodlight logs to determine if any note-worthy errors are ZTN-related. If nothing relevant is found in those logs, proceed to step 4.

4. If possible, investigate the EOS switch using the **`connect switch switch-name`** command. The following is an example of a properly configured and running switch. If Zerotouch state does not indicate **`Zerotouch handshake is complete`** this might point to a serious and non-trivial issue with the ZTN process between the controller and switch.

```
c1(config)# connect switch fixed-eos
fixed-eos> en
fixed-eos(config)# show management dmf controller zerotouch

ZTN is active
Controllers: 10.243.255.239
Manifest timestamp: 2024-06-05 UTC 16:10:37.728336
Zerotouch state: Zerotouch handshake is complete
```

12.20.8.5 Limitations

- The **`Management1`** interface should be active during the switch reboot and image upgrade.
- A default gateway is configured on the switch when the feature is enabled.
- The feature doesn’t work with DHCP configuration on the Management interface.
- The feature only works for fixed system chassis.
- Port configurations such as VLAN tagging, stripping VLAN, rate limit, disabling transmission, and setting truncation size are not allowed for the port chosen as a redundant interface.
- The switch's Management interface can only be configured with CLI commands that are also supported for the loopback interface.
- A LAG/Port-Channel cannot be configured as a redundant management port.
- The front-panel redundant port can have a different MTU value than the Management interface.
- In earlier releases, the CLI command **`show switch all mgmt-stats`** will not display the correct interface redundancy opstate.

12.21 Controller Lockdown

Controller lockdown mode, when enabled, disallows user configuration such as policy configuration, inline configuration, and rebooting of fabric components and disables data path event processing. If there is any change in the data path, it will not be processed.

The primary use case for this feature is a planned management switch upgrade. During a planned management switch upgrade, DANZ Monitoring Fabric (DMF) switches disconnect from the Controller, and DMF policies are reprogrammed, disrupting traffic forwarding to tools. Enabling this feature before starting a management switch upgrade will not disrupt the existing DMF policies when DMF switches disconnect from the Controller, thereby forwarding traffic to the tools.



Note: DMF policies are reprogrammed when the switches reconnect to the DMF fabric when Controller lockdown mode is disabled after the management switch upgrade is completed. Controller lockdown mode is a special operation and should not be enabled for a prolonged period.

- Operations such as switch reboot, Controller reboot, Controller failover, Controller upgrade, policy configuration, etc., are disabled when Controller lockdown mode is enabled.
- The command to enable Controller lockdown mode, **`system control-plane-lockdown enable`**, is not saved to the running config. Hence, Controller lockdown mode is disabled after Controller power down/up. When failover happens with a redundant Controller configured, the new active Controller will be in Controller lockdown mode but may not have all policy information.
- In Controller lockdown mode, copying the running config to a snapshot will not include the **`system control-plane-lockdown enable`** command.
- The CLI prompt will start with the prefix **`LOCKDOWN`** when this feature is enabled.
- Link up/down and other events during Controller lockdown mode are processed after Controller lockdown mode is disabled.
- All the events handled by the switch are processed in Controller lockdown mode. For example, traffic is hashed to other members automatically in Controller lockdown mode if one LAG member fails. Likewise, all switch-handled events related to inline are processed in Controller lockdown mode.

Use the below commands to enable Controller lockdown mode. Only an admin user can enable or disable this feature.

```
Controller# configure
Controller(config)# system control-plane-lockdown enable
Enabling control-plane-lockdown may cause service interruption. Do you want to continue ("y" or
 "yes
" to continue):yes
LOCKDOWN Controller(config)#
```

To disable Controller lockdown mode, use the command below:

```
LOCKDOWN Controller(config)# system control-plane-lockdown disable
Disabling control-plane-lockdown will bring the fabric to normal operation. This may cause some
service interruption during the transition. Do you want to continue ("y" or "yes" to continue):
yes
Controller(config)#
```

12.22 CPU Queue Stats and Debug Counters

Switch Light OS (SWL) switches can now report their CPU queue statistics and debug counters. To view these statistics, use the DANZ Monitoring Fabric (DMF) Controller CLI. DMF exports the statistics to any connected DMF Analytics Node.

The CPU queue statistics provide visibility into the different queues that the switch uses to prioritize packets needing to be processed by the CPU. Higher-priority traffic is assigned to higher-priority queues.

The SWL debug counters, while not strictly limited to packet processing, include information related to the Packet-In Multiplexing Unit (PIMU). The PIMU performs software-based rate limiting and acts as a second layer of protection for the CPU, allowing the switch to prioritize specific traffic.



Note: The feature runs on all SWL switches supported by DMF.

Configuration

These statistics are collected automatically and do not require any additional configuration to enable.

To export statistics, configure a DMF Analytics Node. Please refer to the DMF User Guide for help configuring an Analytics Node.

Show Commands

Showing the CPU Queue Statistics

The following command shows the statistics for the CPU queues on a single switch.

```
controller-1> show switch FILTER-SWITCH-1 queue cpu
# Switch            OF Port Queue ID Type      Tx Packets Tx Bytes  Tx Drops Usage
-|-----------------|-------|--------|---------|----------|---------|--------|--------------------------------|
1 FILTER-SWITCH-1   local   0        multicast 830886     164100990 0        lldp, l3-delivery-arp, tunnel-arp
2 FILTER-SWITCH-1   local   1        multicast 0          0         0        l3-filter-arp, analytics
3 FILTER-SWITCH-1   local   2        multicast 0          0         0
4 FILTER-SWITCH-1   local   3        multicast 0          0         0
5 FILTER-SWITCH-1   local   4        multicast 0          0         0        sflow
6 FILTER-SWITCH-1   local   5        multicast 0          0         0
7 FILTER-SWITCH-1   local   6        multicast 0          0         0        l3-filter-icmp
```

There are a few things to note about this output:

- The CPU's logical port is also known as the **local port**.
- The counter values shown are based on the last time the statistics were cleared.
- Different CPU queues may be used for various types of traffic. The **Usage** column displays the traffic that an individual queue is handling. Not every CPU queue is used.

The **`details`** token can be added to view more information. This includes the absolute (or raw) counter values, the last updated time, and the last cleared time.

Showing the Debug Counters

The following command shows all of the debug counters for a single switch:

```
controller-1> show switch FILTER-SWITCH-1 debug-counters
#  Switch            Name                            Value   Description
--|-----------------|-------------------------------|-------|-----------------------------------------|
1  FILTER-SWITCH-1   arpra.total_in_packets          1183182 Packet-ins recv'd by arpra
2  FILTER-SWITCH-1   debug_counter.register          79      Number of calls to debug_counter_register
3  FILTER-SWITCH-1   debug_counter.unregister        21      Number of calls to debug_counter_unregister
4  FILTER-SWITCH-1   pdua.total_pkt_in_cnt           1183182 Packet-ins recv'd by pdua
5  FILTER-SWITCH-1   pimu.hi.drop                    8       Packets dropped
6  FILTER-SWITCH-1   pimu.hi.forward                 1183182 Packets forwarded
7  FILTER-SWITCH-1   pimu.hi.invoke                  1183190 Rate limiter invoked
8  FILTER-SWITCH-1   sflowa.counter_request          9325983 Counter requests polled by sflowa
9  FILTER-SWITCH-1   sflowa.packet_out               7883772 Sflow datagrams sent by sflowa
10 FILTER-SWITCH-1   sflowa.port_features_update     22      Port features updated by sflowa
11 FILTER-SWITCH-1   sflowa.port_status_notification 428     Port status notif's recv'd by sflowa
```

The counter values shown are based on the last time the statistics were cleared.

Add the **`name`** or the **`ID`** token and a debug counter name or ID to filter the output.

Add the **`details`** token to view more information. This includes the debug counter ID, the absolute (or raw) counter values, the last updated time, and the last cleared time.

Clear Commands

Clearing the Debug Counters

The following command will clear all of the debug counters for a single switch:

```
controller-1# clear statistics debug-counters
```

Clearing all Statistics

To clear both the CPU queue stats and the debug counters for every switch, use the following command:

```
controller-1# clear statistics
```



Note: This command is not only limited to switches. It will clear any clearable statistics for every device.

Analytics Export

The following statistics are automatically exported to a connected Analytics Node:

- CPU queue statistics for every switch.

  Note: This does not include the statistics for queues associated with physical switch interfaces.

- The PIMU-related debug counters. These are debug counters whose name begins with **`pimu`**. No other debug counters are exported.

DMF exports these statistics once every minute.



Note: The exported CPU queue statistics will include port number -2, which refers to the switch CPU's logical port.

12.22.1 Troubleshooting

Use the details with the new show commands to provide more information about the statistics. This information includes timestamps showing statistics collection time and the last time the statistics were cleared.

Use the **`redis-cli`** command to query the Redis server on the Analytics Node from the Bash shell on the DMF Controller to view the statistics successfully exported to the Analytics Node.

The following command queries for the last ten exported debug counters:

```
redis-cli -h analytics-ip -p 6379 LRANGE switch-debug-counters -10 -1
```

Likewise, to query for the last ten exported CPU queue stats:

```
redis-cli -h analytics-ip -p 6379 LRANGE switch-queue-stats -10 -1
```

12.22.2 Limitations

- Only the CPU queue stats are exported to the Analytics Node. Physical interface queue stats are not exported.
- Only the PIMU-related debug counters are exported to the Analytics Node. No other debug counters are exported.
- Only SWL switches are currently supported. EOS switches are not supported.

12.23 Egress Filtering

Egress Filtering is an option to send different traffic to each tool attached to the policy's delivery setting. It provides additional filtering at the delivery ports based on the egress filtering rules specified at the interface.

DANZ Monitoring Fabric (DMF) supports egress filtering on the delivery and recorder node interfaces and supports configuring IPv4 and IPv6 rules on the same interface. Only packets with an IPv4 header are subject to the rules associated with the IPv4 token, while packets with an IPv6 header are only subject to the rules associated with the IPv6 token. If any egress filtering rules are configured on the interface, a default drop rule is applied if no traffic matches the configured rules.

Egress Filtering applies to all switches running SWL OS and EOS DCS-7280R/R2/R3 switches.

12.23.1 Configuring Egress Filtering using the CLI

CLI Configuration

The egress filtering feature is configurable at the interface level. To enable it, run the **egress-filtering** command from the **config-switch-if** submode.

```
dmf-controller(config)# switch DCS-7050SX3-48YC8
dmf-controller(config-switch)# interface ethernet18
dmf-controller(config-switch-if)# egress-filtering
dmf-controller(config-switch-if-egress-filtering)#
```

In the **config-switch-if-egress-filtering** submode, enter the rule's sequence number. The number represents the sequence in which the rules are applied. The lowest sequence number will have the highest priority.



Tip: Leave gaps between the sequence numbers so that new rules can be added in the middle later, if necessary.

```
dmf-controller(config-switch-if-egress-filtering)# 1
allow     Forward traffic matching this rule
drop      Drop traffic matching this rule
```

After the sequence number, specify the action of the rule. It can be either **drop** or**allow**.

```
dmf-controller(config-switch-if-egress-filtering)# 1 allow
any ipv4 ipv6
```

After specifying the action, enter the rule target traffic type: **IPv4**, **IPv6**, or **any**.

```
dmf-controller(config-switch-if-egress-filtering)# 1 allow
any   ipv4  ipv6
```

Any Traffic

The following illustrates a rule to allow all traffic on the interface, in this case, **ethernet18**.

```
dmf-controller(config-switch-if-egress-filtering)# 1 allow any
```

And a rule to drop all traffic on the interface, in this case, **ethernet18**.

```
dmf-controller(config-switch-if-egress-filtering)# 1 drop any
```

IPv4 Traffic

To **allow** or **drop** all IPv4 traffic on an interface, use the following commands:

Drop

```
dmf-controller(config-switch-if-egress-filtering)# 1 drop ipv4
dmf-controller(config-switch-if-egress-filtering-ipv4)#
```

Allow

```
dmf-controller(config-switch-if-egress-filtering)# 1 allow ipv4
dmf-controller(config-switch-if-egress-filtering-ipv4)#
```

The following other options are available in the submode **config-switch-if-egress-filtering-ipv4**. DMF supports the following qualifiers for IPv4 traffic filtering along with the IP address, port, and VLAN ranges.

```
dmf-controller(config-switch-if-egress-filtering-ipv4)#
dscp-value      dst-port        icmp-code       ip-proto        src-port        vlan-range
dst-ip          dst-port-range  icmp-type       src-ip          src-port-range
dst-ip-range    ecn-value       ip-fragment     src-ip-range    vlan
```

The following are examples of using the different options.

```
dmf-controller(config-switch-if-egress-filtering-ipv4)# dst-ip 12.123.123.39
dmf-controller(config-switch-if-egress-filtering-ipv4)# ip-proto 6
dmf-controller(config-switch-if-egress-filtering-ipv4)# dst-port 13
dmf-controller(config-switch-if-egress-filtering-ipv4)# src-port-range min 12 max 23
dmf-controller(config-switch-if-egress-filtering-ipv4)# vlan 45
dmf-controller(config-switch-if-egress-filtering-ipv4)# src-ip 12.123.145.39
dmf-controller(config-switch-if-egress-filtering-ipv4)# dscp-value 23
dmf-controller(config-switch-if-egress-filtering-ipv4)# icmp-type 34
dmf-controller(config-switch-if-egress-filtering-ipv4)# icmp-code 59
dmf-controller(config-switch-if-egress-filtering-ipv4)# ecn-value 2
dmf-controller(config-switch-if-egress-filtering-ipv4)# ip-fragment is-fragment
```

IPv6 Traffic

To **allow** or **drop** all IPv6 traffic on an interface, use the following commands:

Drop

```
dmf-controller(config-switch-if-egress-filtering)# 1 drop ipv6
dmf-controller(config-switch-if-egress-filtering-ipv6)#
```

Allow

```
dmf-controller(config-switch-if-egress-filtering)# 1 allow ipv6
dmf-controller(config-switch-if-egress-filtering-ipv6)#
```

The following options are available in the submode **config-switch-if-egress-filtering-ipv6** for IPv6 traffic filtering.

```
dmf-controller(config-switch-if-egress-filtering-ipv6)#
dscp-value      dst-port        icmp-code       ip-proto        src-port        vlan-range
dst-ip          dst-port-range  icmp-type       src-ip          src-port-range
dst-ip-range    ecn-value       ip-fragment     src-ip-range    vlan
```

Configuring an unsupported qualifier on a switch interface results in the following error message:

```
dmf-controller(config)# show fabric warnings egress-filtering-warning
~~~~~~~~~~~~~~~~~~~~~~~~~ Egress filtering warnings ~~~~~~~~~~~~~~~~~~~~~~~~~
# Switch         IF Name      Warning message

-|--------------|------------|------------------------------------------------------------
-----------------------------|
1 DCS-7050SX3    ethernet18   Rule 1 matching on the following field(s) is not supported on the
 switch: ECN; IP fragment
```

Show Commands

The following show command displays the Egress Filtering enabled device name, the information about the specified rule under the **Entry key** column, and the rule's action under the **Entry value** column. DMF uses the table to communicate the egress filtering rules with the device. The data in this table is primarily intended for debugging and communication purposes.

```
dmf-controller# show switch all table egress-flow-1
# Egress-flow-1 Device name        Entry key
 Entry value
-|-------------|------------------|-----------------------------------------------------------
--------|-------------------------------------------------|
1 0             DCS-7050SX3-48YC8  Priority(1000), Port(13), EthType(2048), Ipv4Src(12.123.123.12)
 Name(__Rule1__), Data([0, 0, 0, 0]), NoDrop()
2 1             DCS-7050SX3-48YC8  Priority(0), Port(13)
 Name(__Rule0__), Data([0, 0, 0, 0]), Drop()
```

If any egress filtering warnings are present, they can be seen by running the **`show fabric warnings egress-filtering-warnings`** command. The output lists the switch name and the interface name on which an egress filtering warning is present, with a detailed message.

Validation Messages

The following are examples of validation failure messages and their potential causes.

A validation exception occurs when configuring an egress filtering rule without specifying **`EtherType`**.

```
Validation failed: EtherType is mandatory for egress filtering rule
```

Similarly, there is another configuration validation for action, which is mandatory for egress filtering rules. Each rule can have a maximum of two ranges, and when exceeded, a validation failure occurs.

```
Validation failed: A rule cannot contain more than 2 configured ranges
```

DMF does not support configuring individual port values and port ranges of the same qualifier in the same rule, and configuring a source port and its range results in a validation failure.

```
Validation failed: Source port and its ranges are not supported together
```

A validation failure occurs if any specified ranges have a higher minimum value than the maximum value. For example, the specified inner VLAN minimum exceeds the maximum value. Similarly, validation failures may occur for ranges.

```
Validation failed: Inner VLAN min cannot be greater than inner VLAN max
```

The **`ip-proto`** setting is mandatory when specifying any port number, source, or destination. Specifying the source or destination port without **`ip-proto`** causes a validation failure.

```
Validation failed: IP protocol number is mandatory for source port
Validation failed: IP protocol number is mandatory for destination port
```

A validation failure occurs if any unsupported IP protocol number is a source or destination port specified without **ip-proto**. DMF only supports TCP(6), UDP(17), and SCTP(132) protocol numbers for port qualifiers.

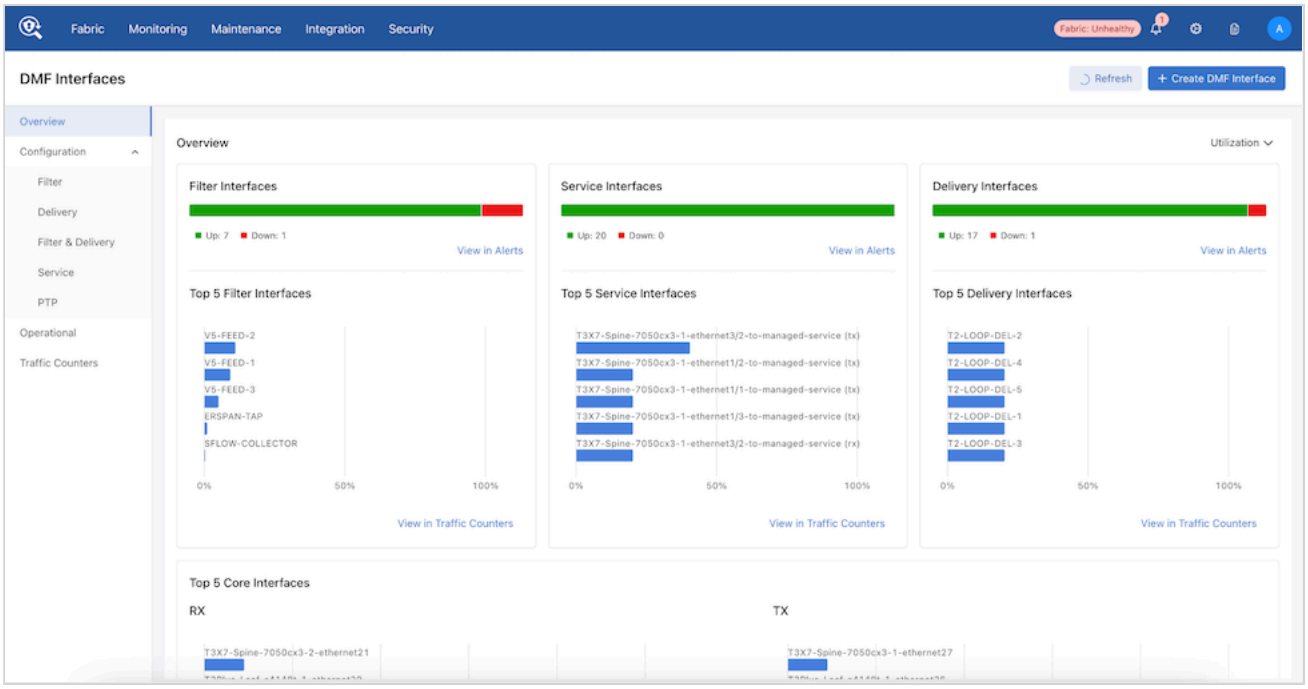
```
Validation failed: IP protocol number protocol number is unsupported for source port
Validation failed: IP protocol number protocol number is unsupported for destination port
```

12.23.2 Configuring Egress Filtering using the GUI

Perform the following steps to configure Egress Filtering.

1. Navigate to **Monitoring** > **Interfaces**.

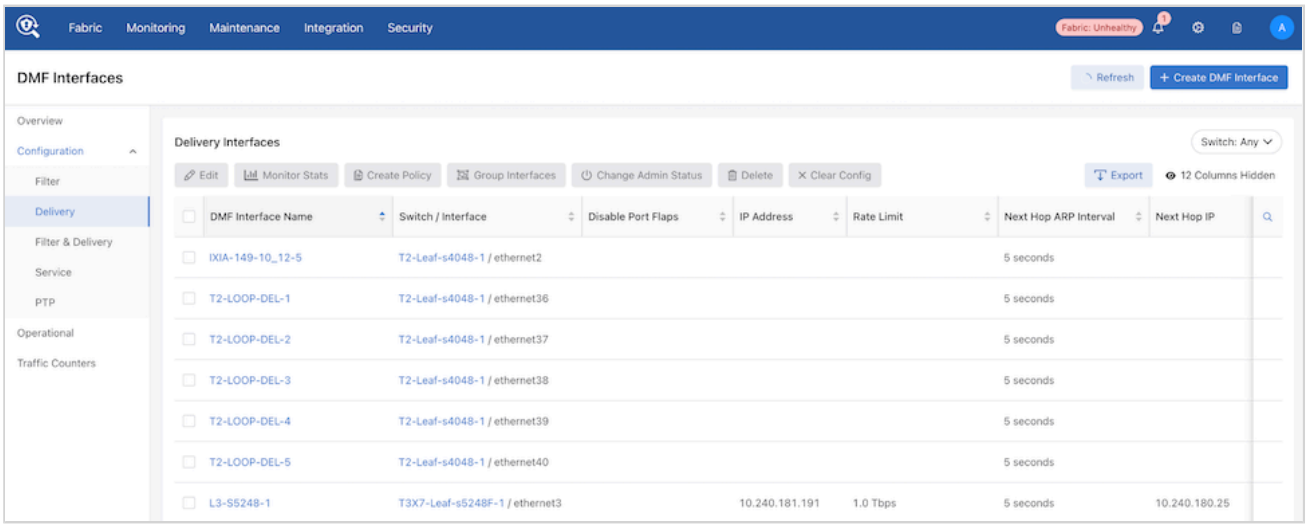
Figure 12-75: DMF Interfaces



2. Under the **Configuration** tab select either **Delivery** or **Filter & Delivery**.

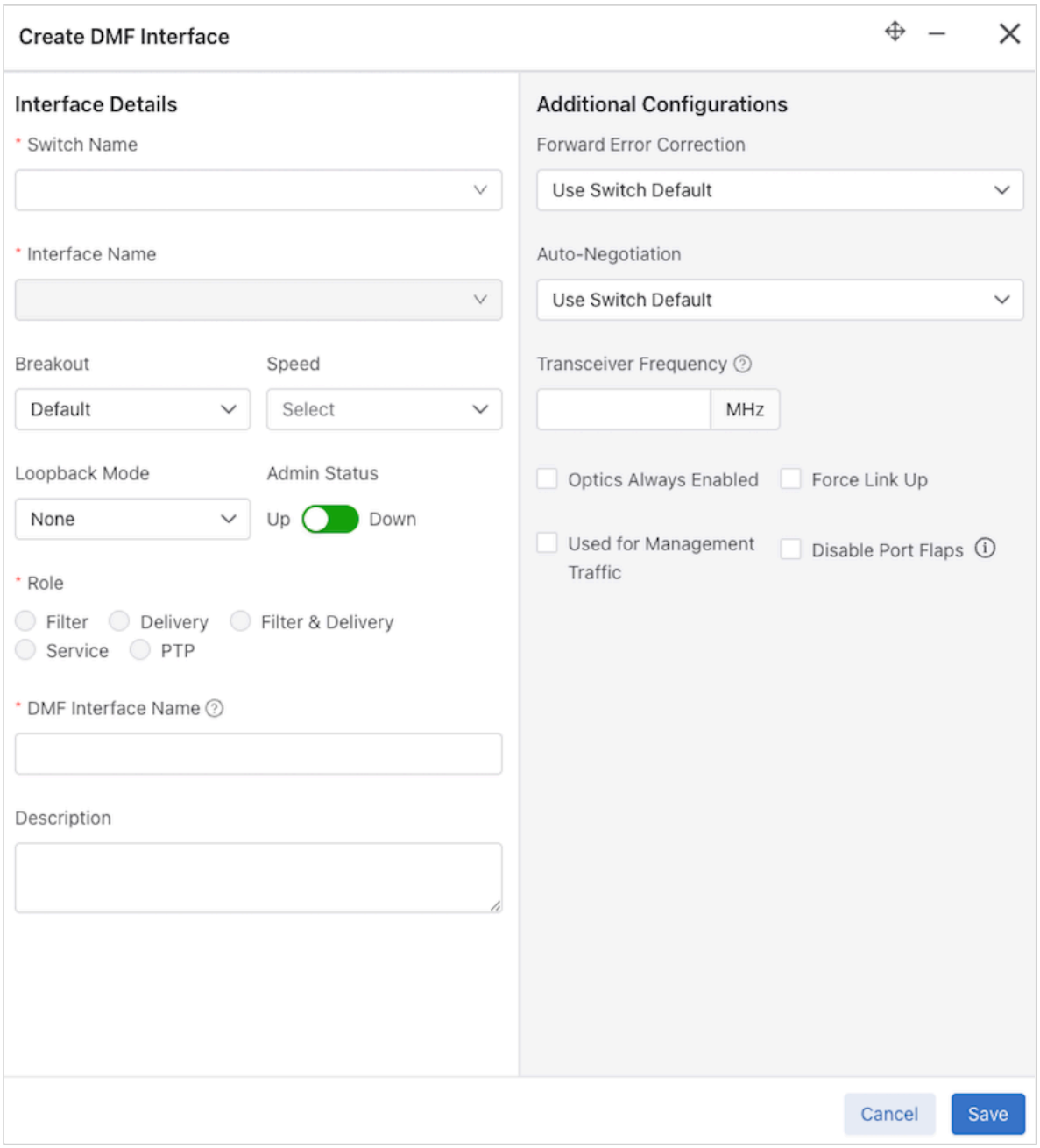
> **Note:** DMF supports configuring egress filtering rules for **Delivery**, **Filter & Delivery**, and **Recorder Node** interfaces.

Figure 12-76: Delivery Interfaces



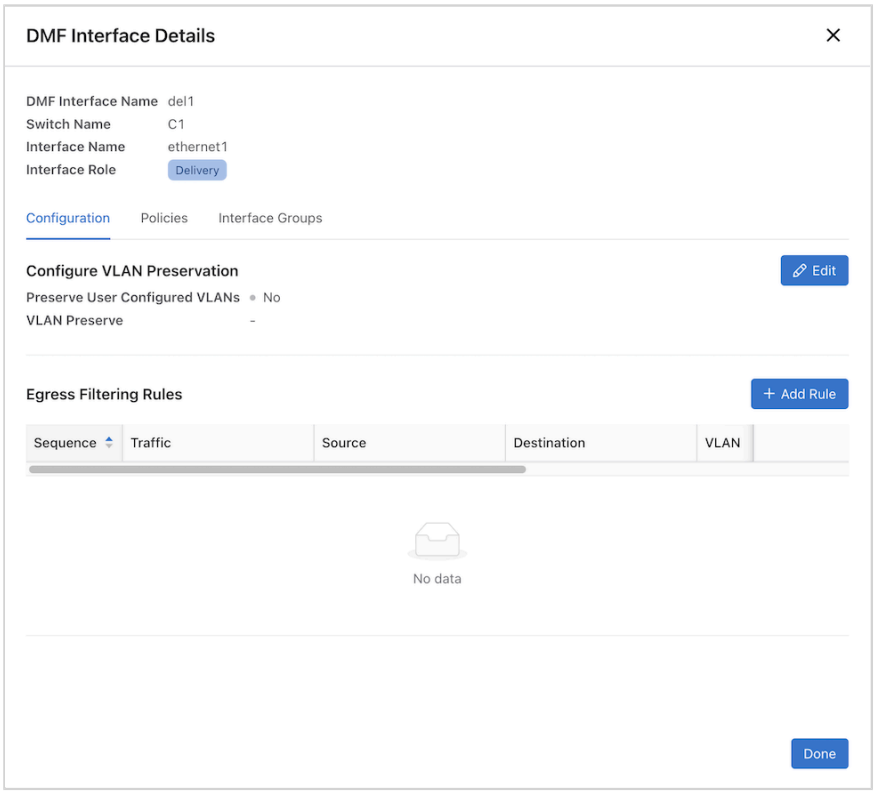
3. **Optional** - If required, create a **Delivery** or **Filter & Delivery** interface using **Create DMF Interface**.

Figure 12-77: Create DMF Interface



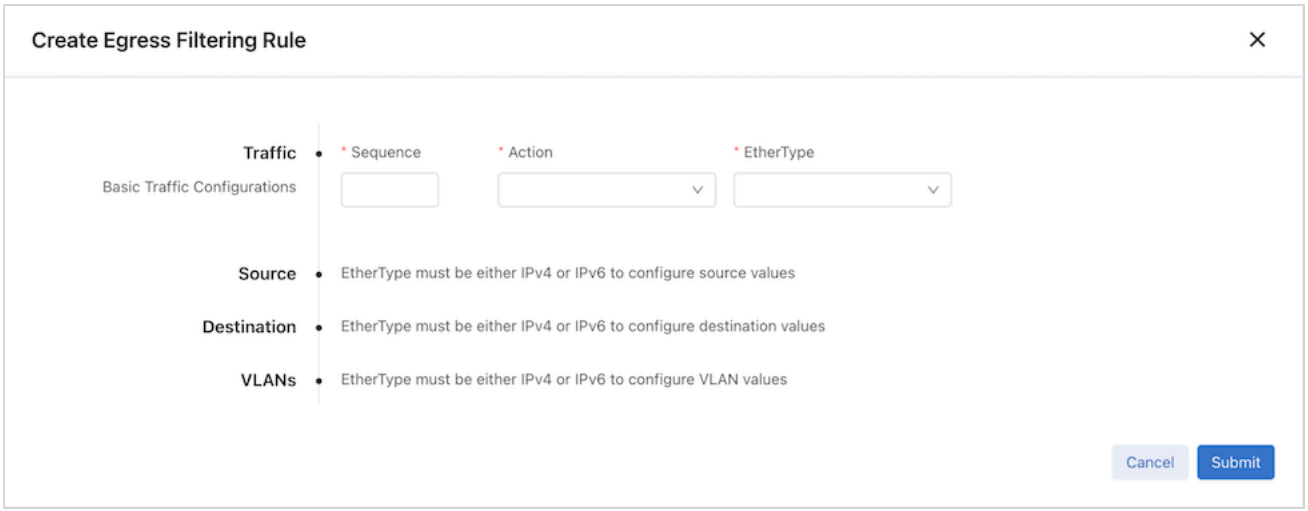
4. Under the **DMF Interface Name**, select the Delivery or Filter & Delivery interface to configure additional interface attributes.

Figure 12-78: DMF Interface Details



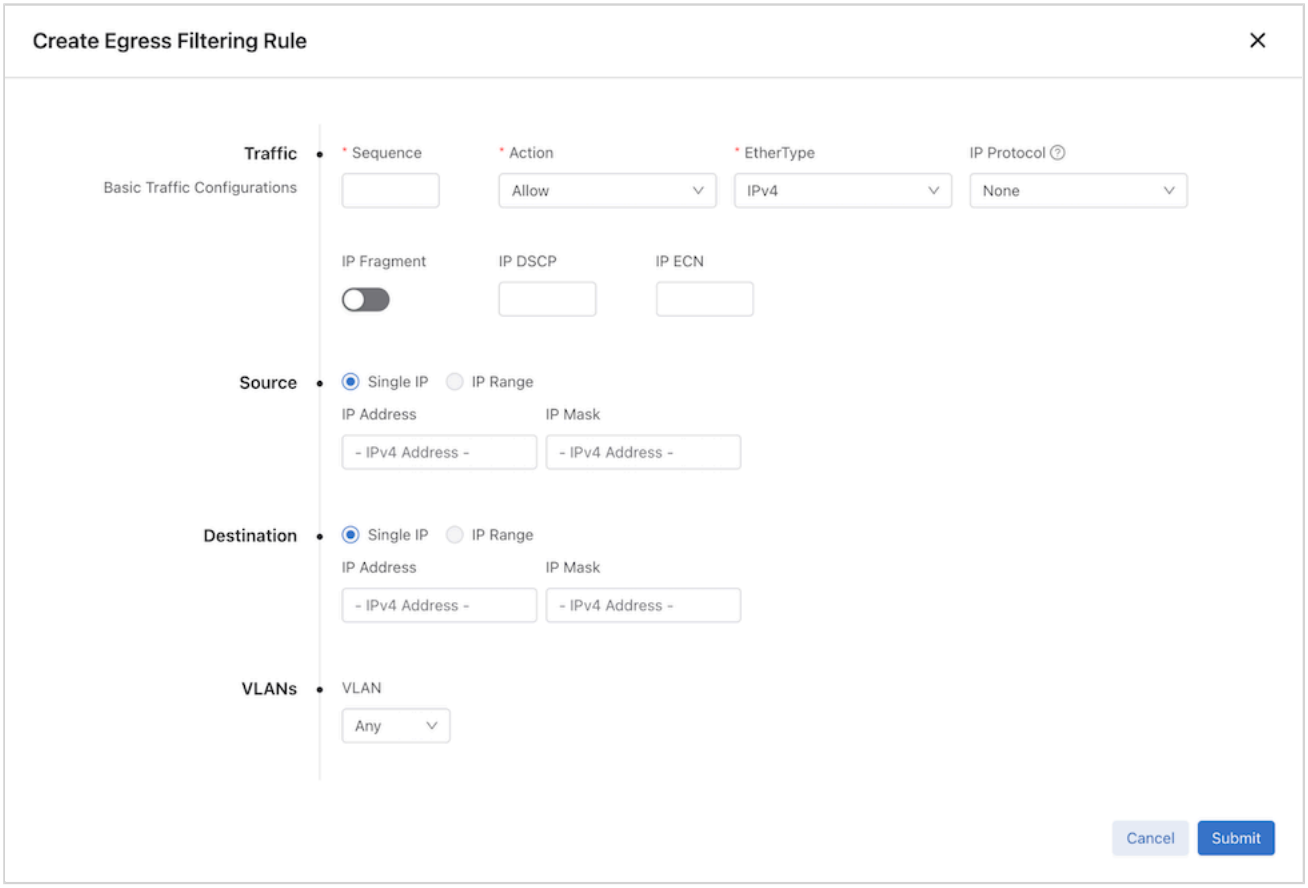
5. Under **Configuration**, select **+ Add Rule** in the **Egress Filtering Rules** section.

Figure 12-79: Add New Rule



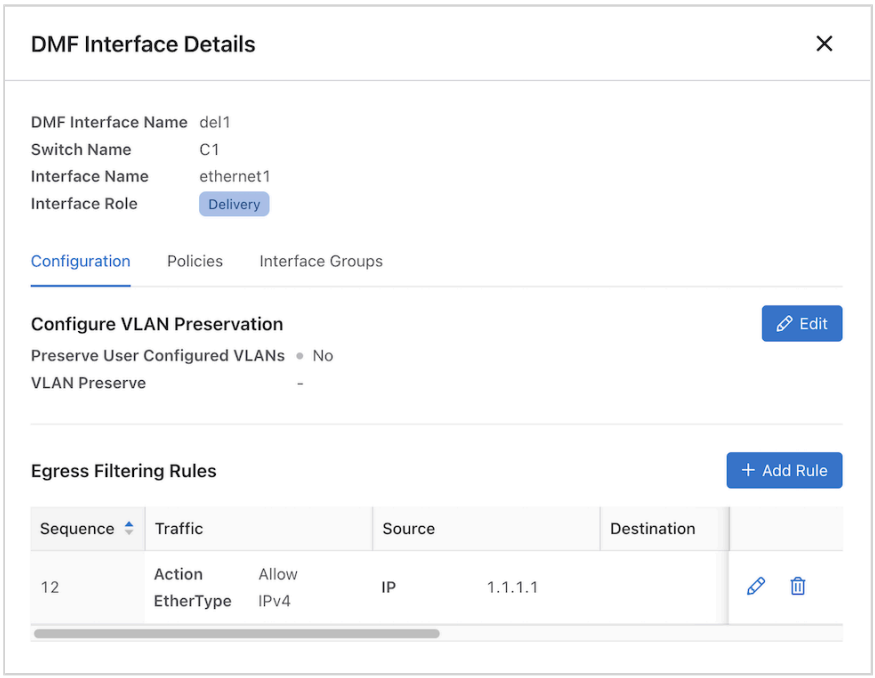
6. Setting **EtherType** to either **IPv4** or **IPv6** displays additional configuration options.
   - **Traffic**: Basic traffic configurations.
     - **Sequence** - Enter a numerical value. The lowest sequence number will have the highest priority.
     - **Action** (drop-down) - **Allow** or **Drop**
     - **Ethertype** (drop-down) - **IPv4**, **IPv6**, or **Any**
   - **Source**: **Single IP** or **IP Range**.
     - IP Address
     - IP Mask
   - **Destination**: **Single IP** or **IP Range**.
     - IP Address
     - IP Mask
   - **VLANs**: **VLAN** (drop-down) - **Any**, **Single**, or **Range**.

Figure 12-80: EtherType Settings



7. Enter all required inputs, and any optional fields that are applicable. Select **Submit** to save the Egress Filtering Rule.
8. Repeat the process using **+ Add Rule** to add more rules as needed.

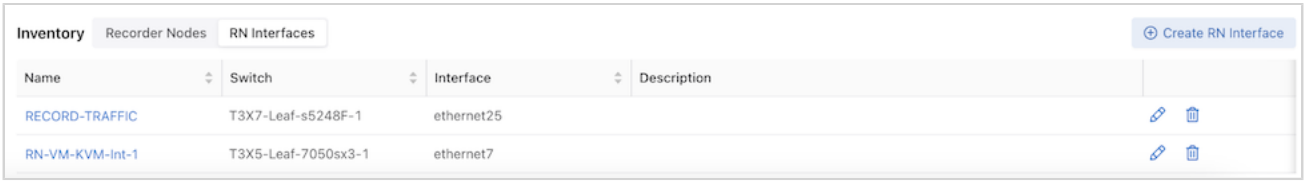
Figure 12-81: Edit Interface



9. To edit a rule, click the **pencil** (edit) icon. To delete a rule, select the **trash** (delete) icon.

Similarly, to configure Egress filtering rules for Recorder Node Interfaces, navigate to **Monitoring** > **Recorder Nodes**. Under the **Inventory** section, select **RN Interfaces** and the RN interface name to configure the egress filtering rules.

Figure 12-82: Inventory



12.23.3 Syslog Messages

There are no Syslog messages relevant to the Egress Filtering feature.

12.23.4 Troubleshooting

When a tool connected to a delivery interface configured with egress filtering rules receives an unexpected packet or does not receive the expected packet, use the following steps to troubleshoot the issue.

1. Review the **`show running-config`** command output to see if the egress filtering rules are configured correctly under that particular interface.
2. Verify the **`show switch`** ***`switch-name`*** **`table egress-flow-1`** command output. It will display the port number of the interface for the configured egress filtering rules, its qualifiers as **`Entry key`**, the **`Entry value`** action of **`Drop`** or **`NoDrop`**, and a default drop rule for that port number with priority **`0`**.

```
dmf-controller# show sw all table egress-flow-1
# Egress-flow-1 Device name        Entry key
    Entry value
-|-------------|-----------------|------------------------------------------------------------
-------|---------------------------------------------|
1 0             DCS-7050SX3-48YC8 Priority(1000), Port(13), EthType(2048), Ipv4Src(12.12
3.123.12) Name(__Rule1__), Data([0, 0, 0, 0]), NoDrop()
2 1             DCS-7050SX3-48YC8 Priority(0), Port(13)
   Name(__Rule0__), Data([0, 0, 0, 0]), Drop()
```

3. Use the following command to verify the same information from a switch (e.g., DCS-7050SX3-48YC8).

```
root@DCS-7050SX3-48YC8:~# ofad-ctl gt egr_flow1
GENTABLE : egr_flow1
GENTABLE ID : 0x0019
Table count: matched/lookup : 0/0
Entry count/limit : 2/1024
guaranteed max: 512, potential max: 1024
priority 0 out_port 13  drop true  0p/0b eid 17
priority 1000 out_port 13 eth_type 0x800/0xffff ipv4_src 12.123.123.12/255.255.255.255  drop
 false  0p/0b eid 22
```

4. Use the **`show fabric warnings egress-filtering-warning`** command to view any egress filtering warnings.

```
dmf-controller(config)# show fabric warnings egress-filtering-warning
~~~~~~~~~~~~~~~~~~~~~~~~~ Egress filtering warnings ~~~~~~~~~~~~~~~~~~~~~~~~~
# Switch               IF Name     Warning message
-|-----------------|-----------|-----------------------------------------------------|
1 DCS-7050SX3-48YC8 ethernet18  Supported only on delivery or recorder node interfaces
2 DCS-7280SR-48C6   Ethernet8   Egress filtering feature is not supported on EOS switches
```

5. The **`show fabric warnings feature-unsupported-on-device`** command provides information on any egress filtering rules configured on any unsupported devices:

```
dmf-controller(config)# show fabric warnings feature-unsupported-on-device
# Name    Warning
-|-------|-----------------------------------------------------------|
1 test1   Egress filtering is not supported on the switch test1
```

12.23.5 Limitations

- Egress filtering supports only 500 rules per interface. A validation failure occurs when exceeding this limit.

```
Validation failed: Only 500 egress filtering rules are supported per interface
```

- DMF does not support egress filtering on MLAG delivery interfaces.
- EOS DCS-7280R/R2/R3 switches do not support **VLAN** qualifiers and their ranges.
- SWL OS switches do not support the following qualifiers: **icmp-code**, **icmp-type**, **ecn-value**, **dscp-value**, and **ip-fragment**.
- For EOS 7280R3, **dst-port** only works on VLAN-tagged packets when there is an exact match.
- None of the IPv6 qualifiers work for EOS 7280R and 7280R2 switches.
- SWL OS switches do not support the following qualifiers under **IPv6** submode: **src-ip**, **dst-ip**, **src-ip-range**, and **dst-ip-range**.

Chapter 13

Advanced Policy Configuration

This chapter describes advanced features and use cases for DANZ Monitoring Fabric (DMF) policies.

13.1 Advanced Match Rules

Optional parameters of a match rule (such as **src-ip**, **dst-ip**, **src-port**, **dst-port**) must be listed in a specific order. To determine the permitted order for optional keywords, use the tab key to display completion options. Keywords in a match rule not entered in the correct order results in the following message:

```
Error: Unexpected additional arguments ...
```

13.1.1 Match Fields and Criteria

The following summarizes the different match criteria available:

- **src-ip**, **dst-ip**, **src-mac**, and **dst-mac** are maskable. If the mask for **src-ip**, **dst-ip**, **src-mac**, or **dst-mac** is not specified, it is assumed to be an exact match.
- For **src-ip** and **dst-ip**, specify the mask in either CIDR notation (for example, /24) or dotted-decimal notation (***255.255.255.0***).
- For **src-ip** and **dst-ip**, the mask must be contiguous. For example, a mask of ***255.0.0.255*** or ***0.0.255.255*** is not supported.
- For TCP, the **tcp-flags** option allows a match on the following TCP flags: **URG**, **ACK**, **PSH**, **RST**, **SYN**, and **FIN**.

The following match combinations are not allowed in the same match rule in the same DMF policy.

- **src-ip-range** and **dst-ip-range**
- **src-ip address group** and **dst-ip address group**
- **ip-range** and **ip address group**

DANZ Monitoring Fabric (DMF) supports matching on user-defined L3/L4 offsets instead of matching on these criteria. However, it is not possible to use both matching packet methods in the same DMF. Switching between these match modes may cause policies defined under the previous mode to fail.

Apply match rules to the following fields in the packet header:

```
dscp-value Match on DSCP value. Value range is 0..63
dst-ip Match dst ip
dst-port Match dst port
is-fragment Match if the packet is IP fragmented
is-not-fragment Match if the packet is not IP fragmented
l3-offset Match on l3 offset
l4-offset Match on l4 offset
range-dst-ip Match dst-ip range
range-dst-port Match dst port ramge
range-src-ip Match src-ip range
range-src-port Match src port range
src-ip Match src ip
src-port Match src port
untagged Untagged (no vlan tag)
vlan-id Match vlan-id
vlan-id-range Match vlan-id range
<ip-proto> IP Protocol
```



Warning: Matching on untagged packets cannot be applied to DMF policies when in ***push-per-policy*** mode.

DMF uses a logical AND if a policy match rule has multiple fields. For example, the following rule matches if the packet has **src-ip *1.1.1.1*** AND **dst-ip *2.2.2.2***:

```
1 match ip src-ip 1.1.1.1 255.255.255.255 dst-ip 2.2.2.2 255.255.255.255
```

DMF uses a logical OR when configuring two different match rules. For example, the following matches if the packet has **src-ip *1.1.1.1*** OR **dst-ip *2.2.2.2***:

```
1 match ip src-ip 1.1.1.1 255.255.255.255
2 match ip dst-ip 2.2.2.2 255.255.255.255
```

A match rule with the **any** keyword matches all traffic entering the filter interfaces in a policy:

```
controller-1(config)# policy dmf-policy-1
controller-1(config-policy)# 10 match any
```

The following commands match on the **TCP SYN** and **SYN ACK** flags:

```
1 match tcp tcp-flags 2 2
2 match tcp tcp-flags 18 18
```



Note: In the DMF GUI, when configuring a match on TCP flags, the current GUI workflow also sets the hex value of the TCP flags for the mask attribute. When configuring a different value for the tcp-flags and tcp-flags-mask attributes in a rule via the DMF CLI, editing the rule in the GUI will override the tcp-flags-mask.

13.1.2 Match-except Rules

The following summarizes match-except rules with examples which allow a policy to permit packets that meet the match criteria, except packets that match the value specified using the **except** command.

- Match-except only supports IPv4 source-IP and IPv4 destination-IP match fields.
  - Example - Permit src-ip network, except ip-address:

```
1 match ip src-ip 172.16.0.0/16 except-src-ip 172.16.0.1
```

  - Example - Permit dst-ip network, except subnet

```
1 match ip dst-ip 172.16.0.0/16 except-dst-ip 172.16.128.0/17
```

- In a rule, the **except** condition can only be used with either src-ip or dst-ip, but not with src-ip and dst-ip together.
  - Example - Except being used with src-ip:

```
1 match icmp src-ip 172.16.0.0/16 except-src-ip 172.16.0.1 dst-ip 172.16.0.0/16
```

  - Example - Except being used with dst-ip:

```
1 match icmp src-ip 224.248.0.0/24 dst-ip 172.16.0.0/16 except-dst-ip 172.16.0.0/18
```

- Except-src-ip or except-dst-ip can only be used after a match for src-ip or dst-ip, respectively.
  - Example - Incorrect match rule:

```
1 match icmp except-src-ip 192.168.1.10
```

  - Example - Correct match rule:

```
1 match icmp src-ip 192.168.1.0/24 except-src-ip 192.168.1.10
```

- In a match rule, only one IP address, or one subnet (range of IP addresses) can be used with the **except** command.
  - Example - Deny a subnet:

```
1 match ip dst-ip 172.16.0.0/16 except-dst-ip 172.16.0.0/18
```

  - Example - Deny an IP Address:

```
1 match ip dst-ip 172.16.0.0/16 except-dst-ip 172.16.0.1
```

13.1.3 Matching with IPv6 Addresses

The value of the EtherType field determines whether the src-ip field to match is IPv4 or IPv6. The DANZ Monitoring Fabric (DMF) Controller displays an error if there is a mismatch between the EtherType and the IP address format.

DMF supports IPv6 address/mask matching, either on src-IP or dst-IP. Optionally, UDP/TCP ports can be used with the IPv6 address/mask match. Specify an address/mask or a group; DMF does not support ranges for IPv6 addresses.



Note: Match rules containing both MAC addresses and IPv6 addresses are not accepted and cause a validation error.

- The preferred IPv6 address representation is as follows: ***xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx***, where each ***x*** is a hexadecimal digit representing ***4*** bits.
- IPv6 addresses range from ***0000:0000:0000:0000:0000:0000:0000:0000*** to ***ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff***.

  In addition to this preferred format, IPv6 addresses may be specified in two other shortened formats:

  - Omit Leading Zeros: Specify IPv6 addresses by omitting leading zeros. For example, write IPv6 address ***1050:0000:0000:0000:0005:0600:300c:326b*** as ***1050:0:0:0:5:600:300c:326b***.
  - Double Colon: Specify IPv6 addresses using double colons (::) instead of a series of zeros. For example, write IPv6 address ***ff06:0:0:0:0:0:0:c3*** as ***ff06::c3***. Double colons may be used only once in an IP address.

DMF does not support the IPv4 address embedded in the IPv6 address format. For example, neither ***0:0:0:0:0:0:101.45.75.219*** nor ***::101.45.75.219*** can be used.

Both IPv4 and IPv6 masks must be in CIDR format. For example, ***FFFF:FFFF:FFFF:FFFF:0:0:0:0*** is valid in DMF, but ***FFFF:0:0:FFFF:FFFF:0:0:0:0*** is not a valid mask.

Both the colon-separated hexadecimal representation and the CIDR-style mask format are supported. The following example illustrates the correct format for IPv6 addresses and subnet masks:

```
controller-1(config)# policy dmf-ipv6-policy
controller-1(config-policy)# 10 match ip6 src-ip 2001::0 ffff:ffff:ffff:ffff:0:0:0:0
controller-1(config-policy)# 11 match ip6 dst-ip 2001:db8:122:344::/64
controller-1(config-policy)# filter-interface all
controller-1(config-policy)# action drop
```

13.1.4 Port and VLAN Range Matches

DANZ Monitoring Fabric (DMF) policy supports matching on source and destination port ranges with optimized hardware resource utilization. DMF uses efficient masking algorithms to minimize the number of flow entries in hardware for each VLAN range. For example, a VLAN range of ***12-99*** uses only five flows in hardware.



Note: Use the untagged keyword to match traffic without a VLAN tag.

Provide the IP protocol information when using source and destination port ranges, fully supported for IPv4 and IPv6 for TCP and UDP. These keywords have the following options:

- **range-dst-ip**: Match dst-ip range.
- **range-dst-port**: Match dst port range.
- **range-src-ip**: Match src-ip range.
- **range-src-port**: Match src port range.

Specify either **src-port-range** or **dst-port-range** or both in each match rule, as illustrated in the following example:

```
controller-1(config)# policy ip-port-range-policy
controller-1(config-policy)# 10 match tcp range-src-port 10 100
controller-1(config-policy)# 15 match udp range-dst-port 300 400
controller-1(config-policy)# 20 match tcp range-src-port 10 2000 range-dst-port 400 800
controller-1(config-policy)# 30 match tcp6 range-src-port 8 20
controller-1(config-policy)# 40 match tcp6 range-src-ip 1:2:3:4::/64 range-src-port 10 300
controller-1(config-policy)# filter-interface all
controller-1(config-policy)# delivery-interface all
controller-1(config-policy)# action forward
```

DMF policy supports matches for the VLAN ID range with optimized hardware resource utilization. Combining a VLAN ID range with a source or destination port range is supported, but not using all three ranges in a single match. The following example illustrates a valid use of the VLAN ID range option:

```
controller-1(config)# policy vlan-range-policy
controller-1(config-policy)# 10 match mac vlan-id-range 30 400
controller-1(config-policy)# 20 match full ether-type ip protocol 6 vlan-id-range 1000 3000 srcip
 1.
1.1.1 255.255.255.255 src-port-range 100 500
```

To determine the number of flow entries required for a range, use the **optimized-match** option, as shown in the following example:

```
controller-1(config-policy)# show running-config policy
! policy
policy vlan-range-policy
action forward
delivery-interface TOOL-PORT-1
filter-interface TAP-PORT-1
10 match mac vlan-id-range 12 99
controller-1(config-policy)# show policy vlan-range-policy optimized-match
Optimized Matches :
10 vlan-min 12 vlan-max 15
10 vlan-min 16 vlan-max 31
10 vlan-min 32 vlan-max 63
10 vlan-min 64 vlan-max 95
10 vlan-min 96 vlan-max 99
```

13.1.5 User Defined Filters

Up to eight two-byte user-defined offsets are allowed on each switch. To view the currently defined offsets, select **Monitoring > User Defined Offsets**.

> **Note:** The DANZ Monitoring Fabric (DMF) Controller must be in push-per-policy mode for a user-defined filter to work accurately.

Selecting the **User Defined Offsets** option when the L3-L4 Offset Match switching mode is not enabled, the system displays a message to enable the correct match mode.

After enabling the L3-L4 Offset Match mode and selecting **Monitoring > User Defined Offsets**, DMF displays a table listing the currently defined offsets.

> **Note:** Matching on a user-defined offset is not recommended when forwarding traffic to a tunnel, because some packets may be dropped.

Each offset match has the following four components:

- **Anchor**: Specified from where the user can define the matching criteria. There are three options: a) L3-start: Start of layer 3 header. b) L4-start: Start of layer 4 header. c) Packet-start: Start of the packet from layer 2 header.
- **Offset**: The number of bytes from the specified anchor.
- **Length**: The number of matching bytes, either 2 or 4 bytes.
- **Value**: The matching value of the specified length in hexadecimal, decimal, or IPv4 format.
- **Mask**: The value that is ANDed with the match value.

> **Note:** DMF allows users to combine up to four 4-byte user-defined offsets or up to eight 2-byte offsets to match up to sixteen bytes in the same match condition. In this case, the multiple offset matching conditions in a single match statement will be considered ANDed. For example, to match on eight bytes, in a single match condition, define two user-defined offsets and configure two rules in an AND fashion so that the first rule matches on the first four bytes and the second rule matches on the remaining four bytes.

Configure each switch with a maximum of eight different offsets matching two bytes each, used in a single policy or any combination in different policies. In the example below, the policy matches on a value of ***0x00001000*** at offset ***40*** from the start of the L3-header and a value of ***0x00002000*** at offset ***64*** from the start of the L4-header.

```
controller-1(config-policy)# 1 match udp dst-port 2152 l3-offset 40 length 4 value 0x00001000 mask
0xffffffff l4-offset 64 length 4 value 0x00002000 mask 0xffffffff
```

Enter the **`show user-defined-offset`** command to display the values configured in the user-defined-offset table.

```
controller-1# show user-defined-offset
# Switch         Slot Anchor   Offset Length Policy
-|--------------|----|--------|------|------|-----------------------------------------------------|
1 DMF-FILTER-SW1 0    l4-start 64     2      DMF-UDF-TEST-1, _DMF-UDF-TEST-1_o_SAVE-TO-RECORDER-NODE
2 DMF-FILTER-SW1 1    l4-start 66     2      DMF-UDF-TEST-1, _DMF-UDF-TEST-1_o_SAVE-TO-RECORDER-NODE
3 DMF-FILTER-SW1 2    l3-start 40     2      DMF-UDF-TEST-1, _DMF-UDF-TEST-1_o_SAVE-TO-RECORDER-NODE
4 DMF-FILTER-SW1 3    l3-start 42     2      DMF-UDF-TEST-1, _DMF-UDF-TEST-1_o_SAVE-TO-RECORDER-NODE
controller-1#
```

DMF supports user-defined filtering on Trident 3 switches. The following are the UDF limitations on a Trident 3 switch in comparison to a non-Trident 3 switch:

UDF Features	Non-Trident SWL Switch	Trident 3 SWL Switch	EOS Switches
Total UDF Length	16 bytes	12 bytes	12 bytes
Minimum Chunk Size	2 bytes	2 bytes	2 bytes
Packet Start (Layer 2 Anchor)	8 offsets	2 offsets	6 offsets
Layer 3 Anchor	8 offsets	6 offsets	6 offsets
Layer 4 Anchor	8 offsets	6 offsets	6 offsets
Layer 2 Offset Range	0 - 126 bytes	0 - 62 bytes	0 - 126 bytes
Layer 3 Offset Range	0 - 114 bytes	0 - 112 bytes	0 - 114 bytes
Layer 4 Offset Range	0 - 96 bytes	0 - 112 bytes	0 - 96 bytes



Note: Please refer to the ***DMF Hardware Compatibility List*** for a complete list of supported switches and their corresponding Network ASIC types.

13.2 Using the Filter and Delivery Role with MAC Loopback for a Two-stage Policy

Use the Filter and Delivery role with a MAC (software) loopback to support monitoring as a service. This option uses a two-stage policy to replicate the incoming feed from one or more filter interfaces and send it to multiple intermediate interfaces (one per end customer or organization).

Define policies on the intermediate interface for forwarding to customer-specific tools. These intermediate interfaces must also be assigned the Filter and Delivery role enabled with the MAC loopback option. This method eliminates the need for a physical loopback cable and a second interface, simplifying monitoring deployment as a service.

When multiple user-defined policies with overlapping rules select traffic from the same filter interfaces for forwarding to different delivery interfaces, overlapping policies are automatically generated to replicate the requisite traffic to the delivery interfaces. The number of overlapping policies increases exponentially with the number of user-defined policies.

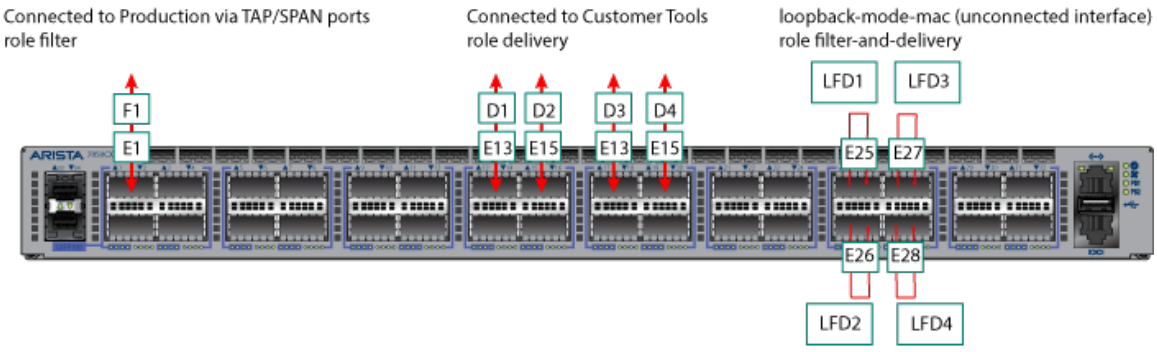
Switch hardware limits limit the total number of policies in the fabric. Using a Filter and Delivery role with a MAC loopback can also help eliminate scale and operational issues seen with overlapping policies.

To configure an interface with the Filter and Delivery role and enable the MAC (software) loopback option, use the **`loopback-mode mac`** command to assign an unused interface as a loopback. This command enables the physical interface without requiring a physical connection to the interface. Use a software loopback interface for copying traffic in any scenario where a physical loopback is required.

The user can also assign the Filter and Delivery role to a software loopback interface, which allows the use of a single interface for copying traffic to multiple destination interfaces. When assigning this role to an interface in loopback mode, use the interface as a delivery interface in relation to the original filter interface and as a filter interface in relation to the final destination interface.

The following figure illustrates the physical configuration for a switch that uses four software loopback interfaces to copy traffic from a single filter interface to four different tools:

Figure 13-1: Using Software Loopback Interfaces to Avoid Overlapping Policies



Use this configuration to copy different types of traffic from a single filter interface (***F1***) to four delivery interfaces (***D1*** to ***D4***). Assign the Filter and Delivery role to the software loopback interfaces (***LFD1*** through ***LFD4***) using just four physical interfaces. Physical loopbacks would require twice as many interfaces.

Considerations

1. The SFP decides the Mac loopback speed. DMF uses the max port speed if there is no SFP (i.e., an empty port).
2. The port speed configuration (if any) will not impact the Mac loopback speed. The Mac loopback speed is set based on the SFP or the max port speed if there is no SFP.
3. The Rate-limit option limits the Mac loopback traffic at Rx side.

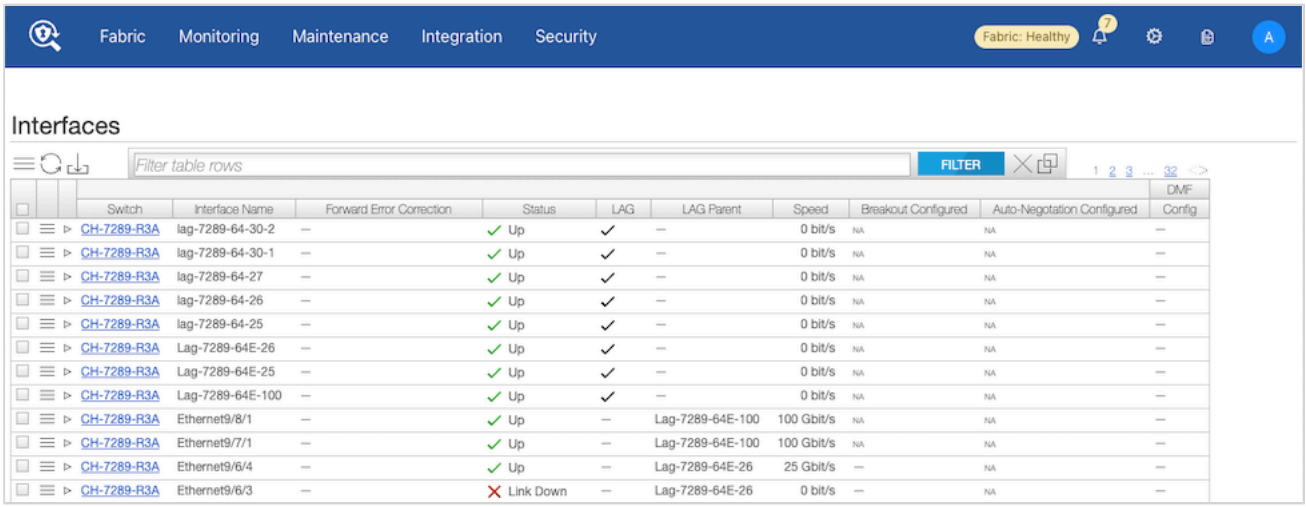
13.2.1 Configure Filter and Delivery Interfaces with MAC Loopback

To configure an interface with the Filter and Delivery role and enable the MAC (software) loopback option, perform the following steps:

1. Display the available interfaces by selecting **Fabric > Interfaces**.

   The system displays the **Interfaces** page, which lists the interfaces connected to the DANZ Monitoring Fabric (DMF) fabric.

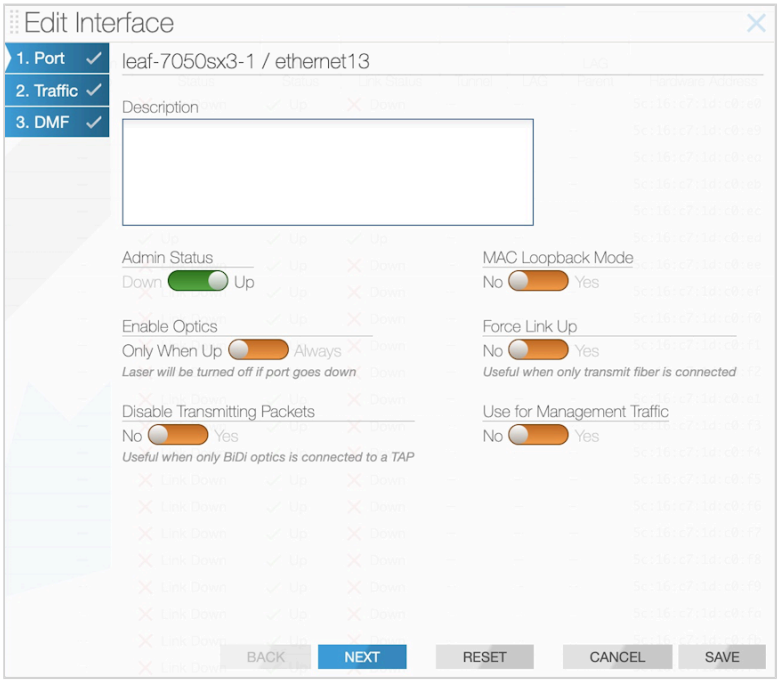
   Figure 13-2: Fabric Interfaces



2. Select the **Menu** control for the interface to use and select **Configure** from the pull-down menu.

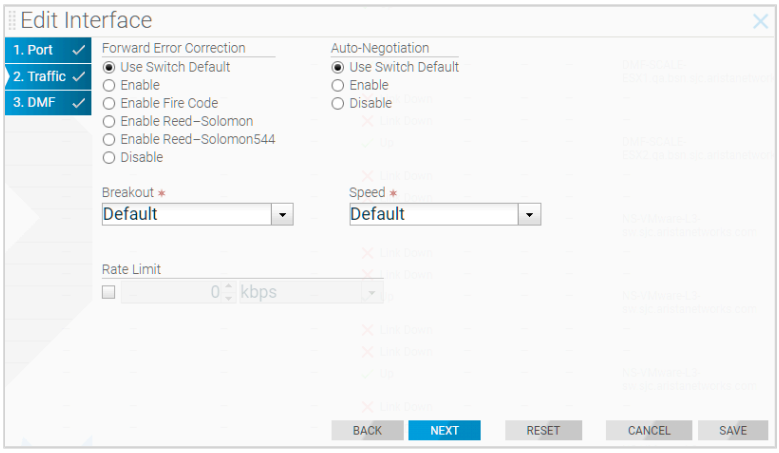
   The system displays the following dialog:

   Figure 13-3: Fabric > Interfaces > Edit Interface > Port



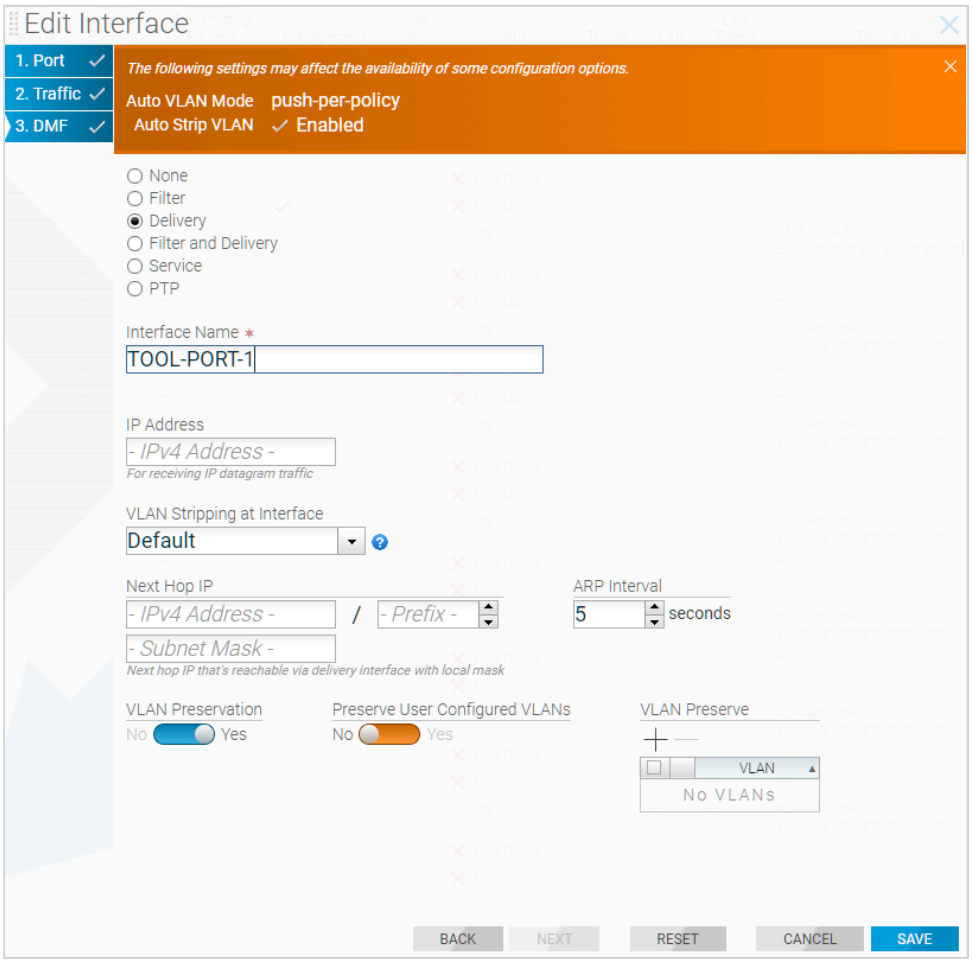
3. (Optional) Type a description for the interface.
4. Enable the **MAC Loopback Mode** slider.
5. Select **Next**.

Figure 13-4: Fabric > Interfaces > Edit Interface > Traffic



6. (Optional) Configure **Rate Limiting**, if required, and select **Next**.

Figure 13-5: Fabric > Interfaces > Edit Interface > DMF



7. Enable the **Filter and Delivery** radio button.
   Optionally enable the **Rewrite VLAN** feature.

   Note: The rewrite VLAN ID feature cannot be used with tunneling.

8. Select **Save** to complete and save the configuration.

13.2.2 Using the CLI To Configure a Filter and Delivery Interface with MAC Loopback

The CLI interface configuration for copying traffic to multiple delivery ports is shown in the following example:

```
switch DMF-FILTER-SWITCH-1
admin hashed-password
$6$5niT1gPm$Jc24qOMF.hxNPI20DvnKaFZKYD6lIo59IMp3O4xIdwVTu2hx0s8Djpvz9xXAXXndiSkKe5jH.9PKoHHrWviSl0
mac 70:72:cf:dc:99:5c
interface ethernet1
role filter interface-name TAP-PORT-1
interface ethernet13
role delivery interface-name TOOL-PORT-1
interface ethernet15
role delivery interface-name TOOL-PORT-1
interface ethernet17
role delivery interface-name TOOL-PORT-3
interface ethernet19
role delivery interface-name TOOL-PORT-4
interface ethernet25
loopback-mode mac
role both-filter-and-delivery interface-name LOOPBACK-PORT-1
interface ethernet26
loopback-mode mac
role both-filter-and-delivery interface-name LOOPBACK-PORT-2
interface ethernet27
loopback-mode mac
role both-filter-and-delivery interface-name LOOPBACK-PORT-3
interface ethernet28
loopback-mode mac
role both-filter-and-delivery interface-name LOOPBACK-PORT-4
```

The following example illustrates using five policies to implement this use case without creating overlapping policies. Otherwise, sixteen overlapping policies would be created without using the loopback interfaces to copy the traffic to separate filter interfaces.

```
! policy
policy TAP-NETWORK-1
action forward
delivery-interface LOOPBACK-PORT-1
delivery-interface LOOPBACK-PORT-2
delivery-interface LOOPBACK-PORT-3
delivery-interface LOOPBACK-PORT-4
filter-interface TAP-PORT-1
1 match any
!
policy DUPLICATED-TRAFFIC-1
action forward
delivery-interface TOOL-PORT-1
filter-interface LOOPBACK-PORT-1
1 match ip src-ip 100.1.1.1 255.255.255.252
!
policy DUPLICATED-TRAFFIC-2
action forward
delivery-interface TOOL-PORT-2
filter-interface LOOPBACK-PORT-2
1 match ip dst-ip 100.1.1.1 255.255.255.252
!
policy DUPLICATED-TRAFFIC-3
action forward
delivery-interface TOOL-PORT-3
filter-interface LOOPBACK-PORT-3
1 match tcp src-port 1234
!
policy DUPLICATED-TRAFFIC-4
action forward
delivery-interface TOOL-PORT-4
filter-interface LOOPBACK-PORT-4
1 match tcp dst-port 80
```

Use the **`show policy`** command to verify the policy configuration.

13.3 Rate Limiting Traffic to Delivery Interfaces

The option exists to limit the traffic rate on a delivery interface, which can be a regular interface, a port channel, a tunnel interface, or a loopback interface.

For information about using rate limiting on tunnels, refer to the section Using the CLI to Rate Limit the Packets on a VXLAN Tunnel.

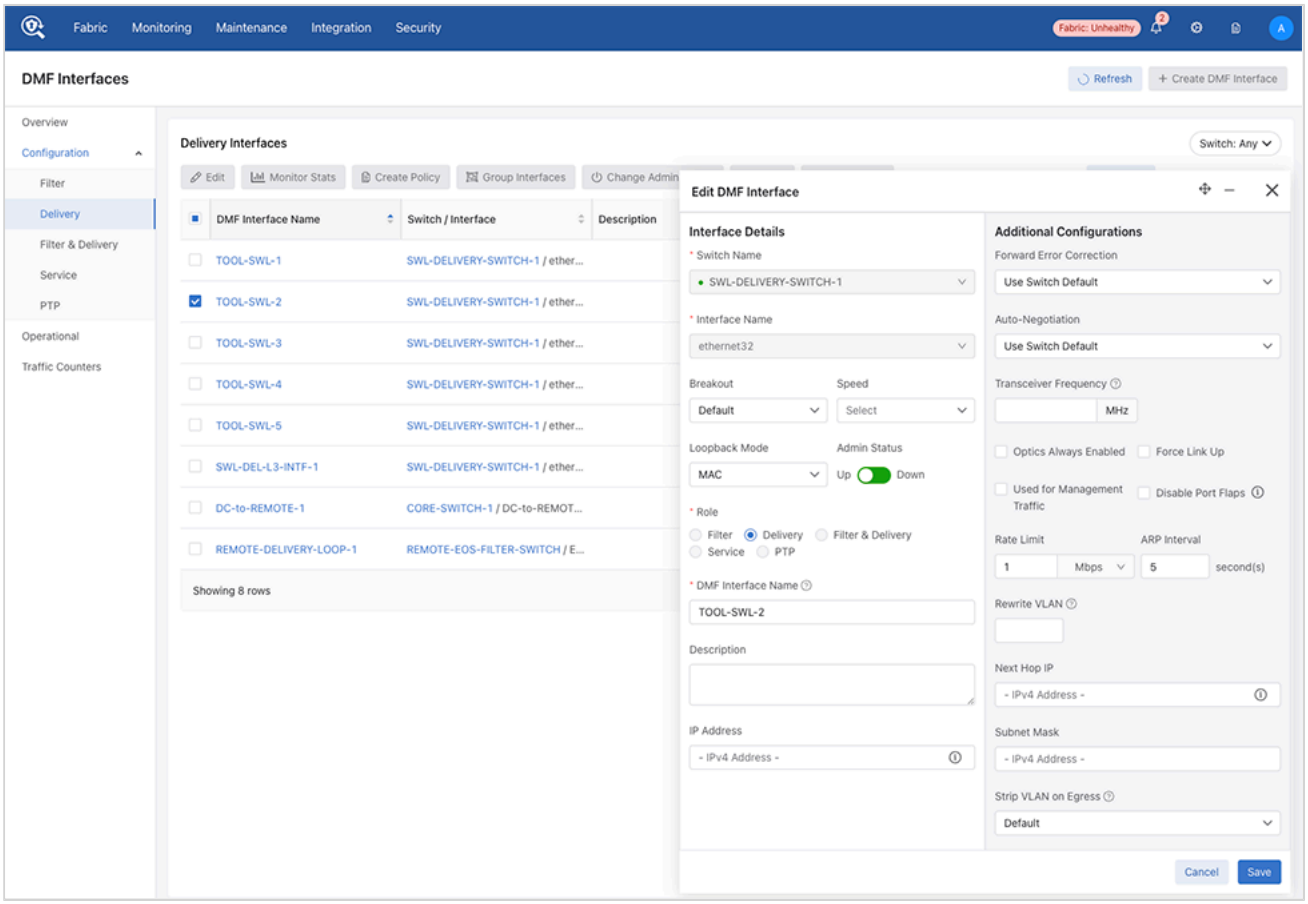
Use kbps to configure the rate-limit for the regular delivery interface. Arista Networks recommends configuring the rate limit in multiples of ***64*** kbps.

13.3.1 Rate Limiting Using the GUI

To use the GUI to set the rate limit for an interface, perform the following steps:

1. Select **Monitoring** > **Interfaces** > **Configuration** > **Delivery**.

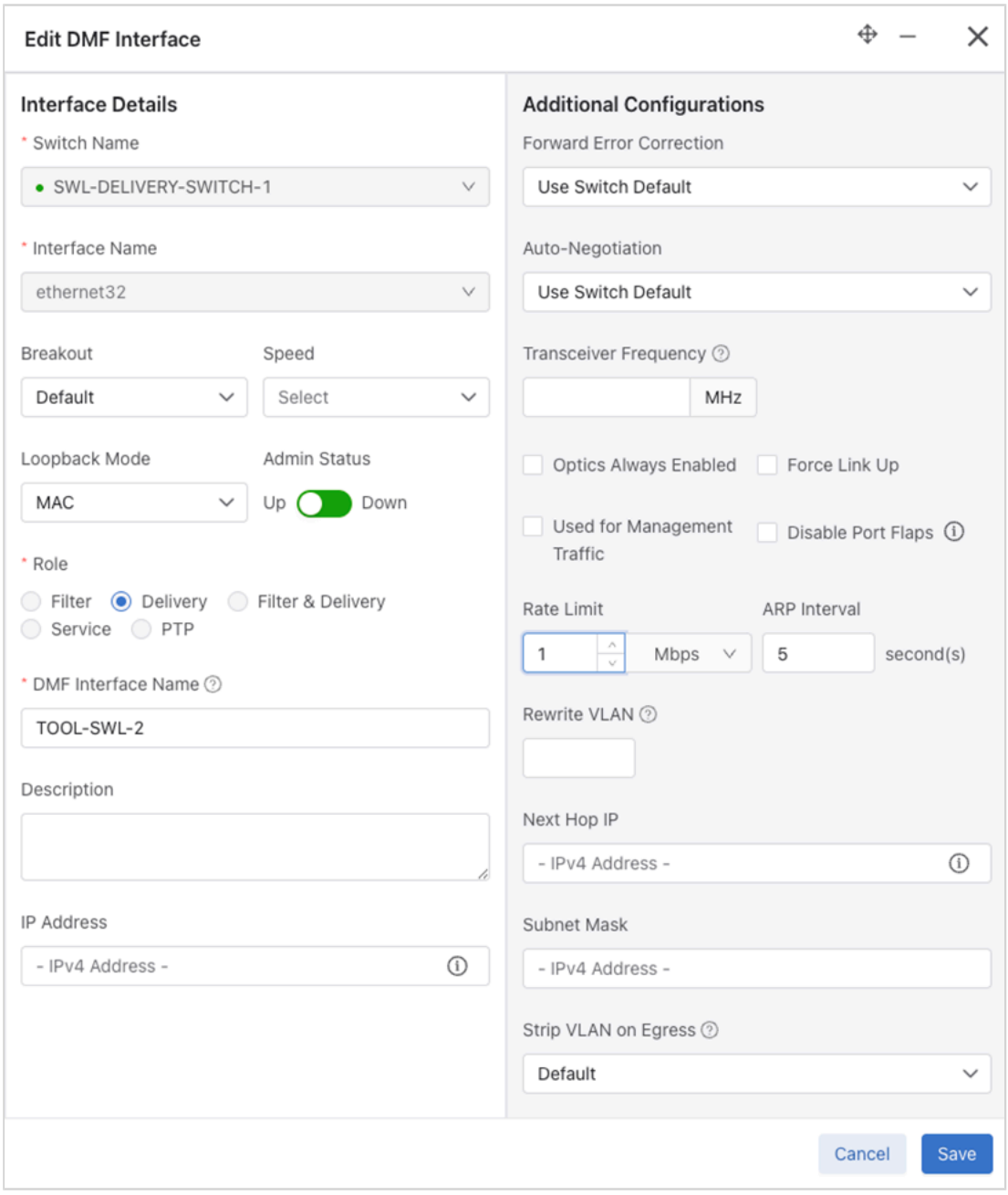
Figure 13-6: Delivery Interfaces



2. Select (checkbox) a specific interface and select **Edit**.

3. Select a **Rate Limit** number and the **Bit Rate** from the drop-down.

Figure 13-7: Setting the Rate Limit for an Interface



4. Select **Save**.

13.3.2 Rate Limiting Using the CLI

CLI Procedure

The following example applies a rate limit of ***10*** Mb/s to the delivery **`interface tobcotDelivery`**:

```
CONTROLLER-1(config)#switch DMF-DELIVERY-SWITCH-1
CONTROLLER-1(config-switch)# interface ethernet1
CONTROLLER-1(config-switch-if)# role delivery interface-name TOOL-PORT-1
CONTROLLER-1(config-switch-if)# rate-limit 10240
```

To view the configuration, enter the **`show this`** command, as in the following example:

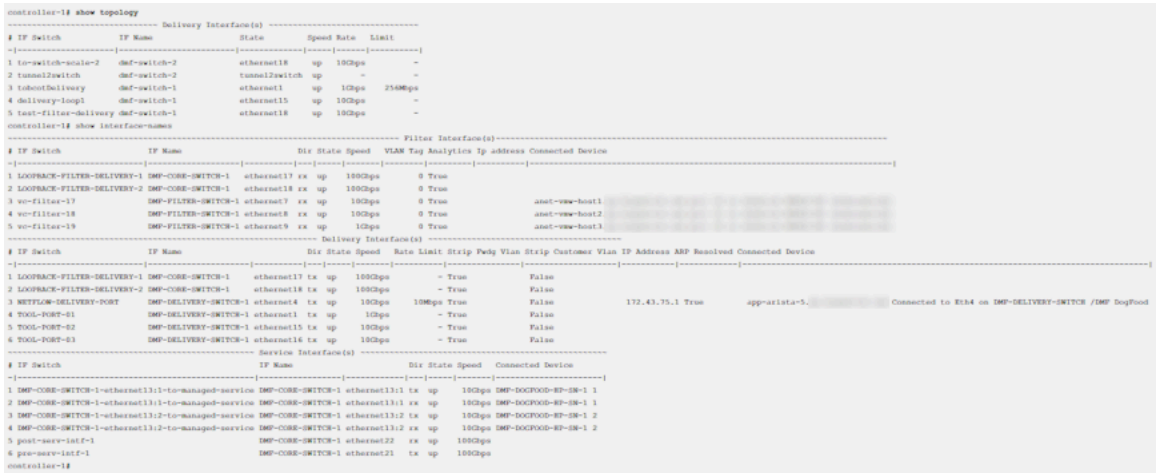
```
CONTROLLER-1(config-switch-if)# show this
! switch
switch DMF-DELIVERY-SWITCH-1
!
interface ethernet1
rate-limit 10000
role delivery interface-name TOOL-PORT-1
CONTROLLER-1 (config-switch-if)#
```

Configure the rate limit for each member interface to rate limit a port channel. Configure individual rate limits for each member interface if the port channel has two member interfaces.

```
lag-interface lag1
hash-type l3
member ethernet43
member ethernet45
interface ethernet43
rate-limit 10000 <------ set the rate-limit to 10 Mbps
interface ethernet45
rate-limit 128000 <---------- set the rate-limit to 128 Mbps
```

To display the configured rate limit, use the **`show topology`** and **`show interface-names`** commands, as in the following examples:

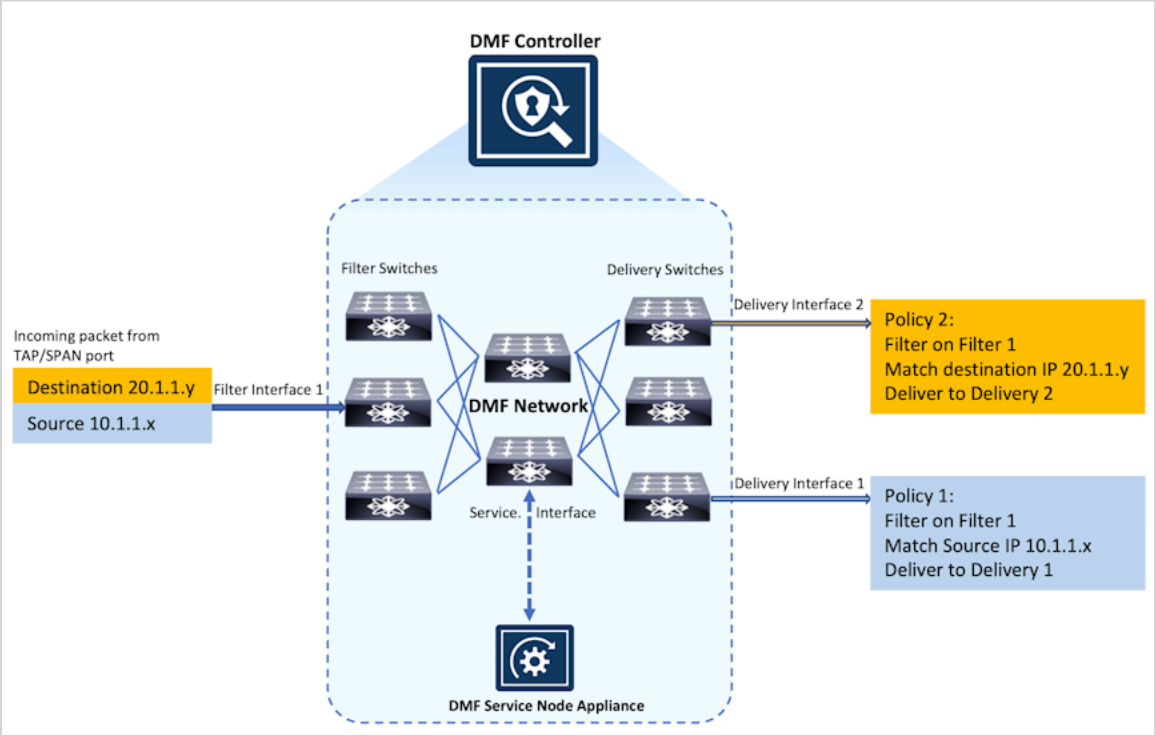
> **Note:** In the current release, the Rate Limit column does not show the configured value for LAG and tunnel interfaces.



13.4 Configuring Overlapping Policies

When two or more policies have one or more filter ports in common, the match rules in these policies may intersect. If the priorities are different, the policy with the higher priority takes effect. However, if the policies have the same priority, the policies overlap, as illustrated in the figure below:

Figure 13-8: Overlapping Policies



In the policy illustrated, packets received on interface ***Filter 1*** with the source-IP address ***10.1.1.x/24*** are delivered to ***D1***. In a separate policy, with the same priority, packets received at ***Filter 1*** with the destination IP address ***20.1.1.y/24*** are delivered to ***D2***. With both these policies applied, when a packet arrives at ***F1*** with a source IP address ***10.1.1.5/24*** and a destination IP address ***20.1.1.5/24***, the packets are copied and forwarded to both ***D1*** and ***D2***. Enabled by default, the DANZ Monitoring Fabric (DMF) policy overlap feature causes this behavior.

DMF manages overlapping policies automatically by copying packets received on the same filter interface that match multiple rules but which the policy forwards to different delivery interfaces.

Two policies are said to be overlapping when all of the following conditions are met:

- At least one delivery interface is different.
- At least one filter interface is shared.
- Match rules across policies intersect, which occurs under these conditions:
  - The match rules match on the same field, but a different value OR both policies have the same configured priority (or same default priority).
  - The match rules match on completely different fields.

Note: Automatically created dynamic policies will be visible in the **`show policy`** command. However, they will not be visible in the **`running config`**, nor can they get deleted manually.

When overlapping policies are detected, by default, DMF performs the following operations:

- Creates a new dynamic policy that aggregates the policy actions.
- Assigns policy names, using this dynamic policy naming convention: _***policy1***_**o**_***policy2***_
- Adds match combinations and configuration as appropriate.
- Assigns a slightly higher priority to the new aggregated policy so that it overrules the overlapping policies, which, as a result, only applies to traffic that does not match the new aggregated policy. An incremental value of .1 is added to the original policy priority. For example, if the original policies have a priority of ***100***, the dynamic policy priority is ***101***.

  Note: When changing the configurable parameters in an existing DMF out-of-band policy, any counters associated with the policy, including service-node-managed services counters, are reset to zero.

The **`overlap-limit-strict`** command, enabled by default, strictly limits the number of overlapping policies to the maximum configured using the **`overlap-policy-limit`** command. For example, the operation fails with a validation error when setting the maximum number of overlapping policies to four (the default) and attempting to create a fifth policy using the same filter interface. To disable strict enforcement, use the **`no overlap-limit-strict`** command.

Note: The **`overlap-strict-limit`** command is disabled and must be manually enabled to enforce configurable policy limits.

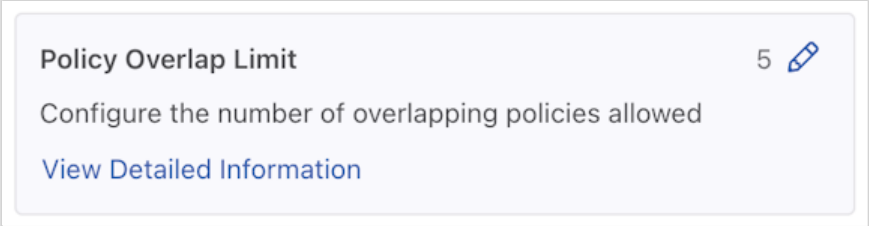
13.4.1 Configuring the Policy Overlap Limit Using the GUI

Policy Overlap Limit

Perform the following steps to configure the Policy Overlap Limit.

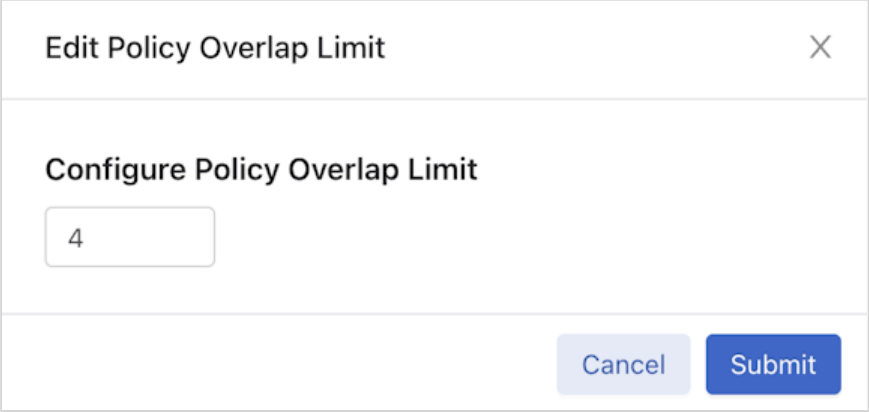
1. Control the configuration of this feature using **Edit** by locating the corresponding card and selecting the **pencil** icon.

   Figure 13-9: Policy Overlap Limit



2. A configuration edit dialogue window pops up, displaying the corresponding prompt message. By default, the Policy Overlap Limit is 4.

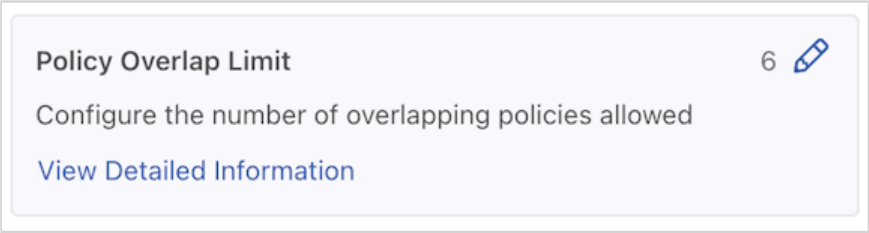
   Figure 13-10: Edit Policy Overlap Limit



3. Adjust the Value (minimum value: 0, maximum value: 10). There are two ways to adjust the value:
   - Directly enter the desired value in the input area.
   - Use the up and down arrow buttons in the input area to adjust the value accordingly. Pressing the up arrow increments the value by 1, while pressing the down arrow decrements it by 1.
4. Select **Submit** to confirm the configuration changes or **Cancel** to discard the changes.

5. After successfully setting the configuration, the current configuration status displays next to the **Pencil** (edit) icon.

Figure 13-11: Policy Overlap Limit Change Success



13.4.2 Configuring the Overlapping Policy Limit Using the CLI

By default, the number of overlapping policies allowed is ***four***. The maximum number to configure for overlapping policies is ***ten***. Set the overlap policy limit to ***zero*** to disable the overlapping policy feature.

To change the default limit for overlapping policies, use the following command:

```
controller-1(config)# overlap-policy-limit integer
```

Replace ***integer*** with the maximum number of overlapping policies to support fabric-wide.

For example, the following command sets the number of overlapping policies supported to the maximum value (***10***):

```
controller-1(config)# overlap-policy-limit 10
```

The following command disables the overlapping policies feature:

```
controller-1(config)# overlap-policy-limit 0
```



Note: When setting the Policy Overlap Limit to *zero*, ensure the policies do not overlap. If active policies overlap after disabling this feature, the forwarding result may be unpredictable.

13.4.3 Using the CLI to View Overlapping Policies

Enter the **show policy** command to view statistics for dynamic (overlapping) policies. If an overlapping policy appears in the output, the parent policies are identified, as in the following example:

```
controller-1(config-policy)# show policy
# Policy Name Config Status        Runtime Status Action  Type       Priority Overlap Priority Rewrite VLAN Filter BW Delivery BW Services
-|-----------|--------------------|--------------|-------|----------|--------|----------------|------------|---------|-----------|--------|
1  _p2_o_p1    active and forwarding installed      forward Dynamic    100      1                0            -         -
2 p1           active and forwarding installed      forward Configured 100      0                0            -         -
3 p2           active and forwarding installed      forward Configured 100      0                0            -         -
```

In this example:

- **show overlap _P1_O_P2**, lists component policies: source ***P1***, ***P2***.
- **show P1**, lists dynamic policies: **overlap _P1_O_P2**.

To view the details for a specific overlapping policy, append the policy name to the **show policy** command, as in the following example:

```
controller-1(config-policy)# show policy _p1_o_p2
Policy Name : _p1_o_p2
Config Status : active and forwarding
Runtime Status : installed
Detailed Status : installed - installed to forward
Action : forward
Priority : 100
Overlap Priority : 1
Description : runtime policy
# of switches with filter interfaces : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces : 0
# of filter interfaces : 1
# of delivery interfaces : 2
# of core interfaces : 4
# of services : 0
# of pre service interfaces : 0
# of post service interfaces : 0
Rewrite VLAN : 0
Total Ingress Rate : -
Total Delivery Rate : -
Total Pre Service Rate : -
Total Post Service Rate : -
Overlapping Policies : none
Component Policies : p2, p1,
Failed Overlap Policy Exceeding Max Rules :
Rewrite valid? : False
Service Names :
Overlap Matches :
1 ether-type 2048 src-ip 10.1.1.1 255.255.255.0 dst-ip 20.1.1.1 255.255.255.0
Strip VLAN : False
Delivery Bandwidth : 20 Gbps
explicitly-scheduled : False
Filter Bandwidth : 10 Gbps
Type : Dynamic
~ Match Rules ~
None.
~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~
# IF Switch IF Name State Dir Packets Bytes Pkt Rate Bit Rate
-|---------|-----------|--------|-----|---|-------|-----|--------|--------|
1 f1 filter-sw-1 s11-eth1 up rx 0 0 0 -
~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~
# IF Switch IF Name State Dir Packets Bytes Pkt Rate Bit Rate
-|---------|-----------|--------|-----|---|-------|-----|--------|--------|
1 d1 filter-sw-2 s12-eth1 up tx 0 0 0 -
2 d2 filter-sw-2 s12-eth2 up tx 0 0 0 -
~ Service(s) ~
None.
~~~~~~~~~~~~~~~~~~~~~~ Core Interface(s) ~~~~~~~~~~~~~~~~~~~~~~
# Switch IF State Dir Packets Bytes Pkt Rate Bit Rate
-|-----------|--------|-----|---|-------|-----|--------|--------|
1 filter-sw-1 s11-eth3 up tx 0 0 0 -
2 core-sw-2 s10-eth1 up rx 0 0 0 -
3 core-sw-2 s10-eth2 up tx 0 0 0 -
~ Failed Path(s) ~
None.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Event History ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Time Event Detail
-|-------------------|---------------------|-----------------------------------------|
1 2014-08-05 22:22:27 start forward pending installation - installed to forward
2 2014-08-05 22:22:27 installation complete installed - installed to forward
```

13.4.4 Configuring the Policy Overlap Limit Strict using the GUI

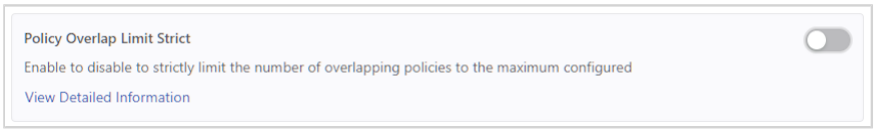
The Policy Overlap Limit Strict option, **enabled** by default, strictly limits the number of overlapping policies to the maximum configured. For example, when setting the maximum number of overlapping policies to 4 (the default) and users create a fifth policy using the same filter interface, the operation fails with a validation error.

From the DANZ Monitoring Fabric (DMF) Features page, proceed to the Configuring the Policy Overlap Limit Strict feature card.

1. Select the **Policy Overlap Limit Strict** card.

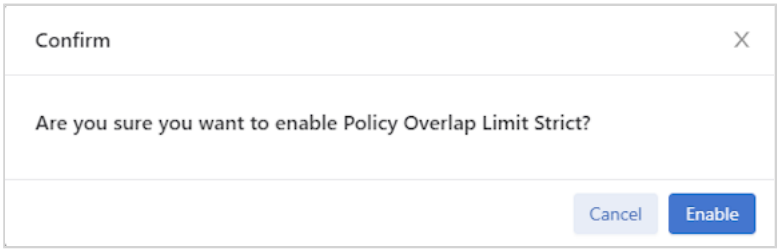
> **Note:** The Policy Overlap Limit Strict option is **enabled** by default. The following steps guide if the Policy Overlap Limit Strict option is disabled.

Figure 13-12: Policy Overlap Limit Strict Disabled



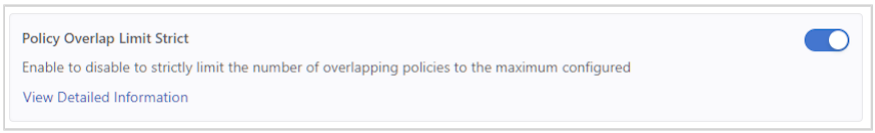
2. Toggle the Policy Overlap Limit Strict switch to **On**.
3. Confirm the activation by selecting **Enable**. Or, **Cancel** to return to the DMF Features page.

Figure 13-13: Enable Policy Overlap Limit Strict



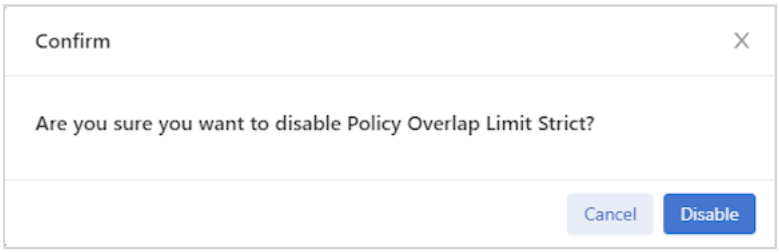
4. Retain Configuring the Policy Overlap Limit Strict is running.

Figure 13-14: Policy Overlap Limit Strict Enabled



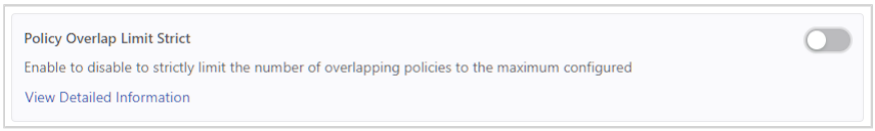
5. To disable the feature, toggle the Policy Overlap Limit Strict switch to **Off**. Select **Disable** and confirm.

Figure 13-15: Disable Policy Overlap Limit Strict



The feature card updates with the status.

Figure 13-16: Policy Overlap Limit Strict Disabled



13.4.5 Configuring the Policy Overlap Limit Strict using the CLI

The Policy Overlap Limit Strict option, **enabled** by default, strictly limits the number of overlapping policies to the maximum configured. For example, when setting the maximum number of overlapping policies to 4 (the default) and users create a fifth policy using the same filter interface, the operation fails with a validation error.

Use the following commands to disable or enable the Policy Overlap Limit Strict feature using the CLI.

```
controller-1(config)# no overlap-limit-strict
```

```
controller-1(config)# overlap-limit-strict
```

13.4.6 Exclude Inactive Policies from Overlap Limit Calculation

Previously, DANZ Monitoring Fabric (DMF) calculated the overlap policy limit by determining how many policies use the same filter interface, irrespective of whether the policies are active or inactive. By default, the overlap policy limit is 4, and the maximum is 10.

Suppose the limit is 4, and a user attempts to create a 5th policy using the same filter interface (`f1`). DMF throws the following error message: `Error: Validation failed: Filter interfaces used in more than 4 policies: f1`.

This count will include filter interfaces used in **`active`** or **`inactive`** policies.

The DMF policy overlap calculation excludes inactive policies when using the **`inactive`** command in policy configuration. For example, using the same policy limit settings described above, DMF supports creating a 5th policy using the same filter interface (**`f1`**) by first putting the 5th policy in an inactive state using the **`inactive`** command under policy configuration.



Note: This feature applies to switches running SWL OS and EOS.

Global Configuration Example

1. Select a switch and enter the config mode using the following command:

```
(config)# switch core1
```

2. Select an interface on the switch used as the filter-interface, as shown in the following example.

```
(config-switch)# interface ethernet1
```

3. Create a filter interface, for example, **`f1`**, using the following command:

```
(config-switch-if)# role filter interface-name f1
```

4. Repeat the process to create the delivery interfaces.

```
(config-switch)# interface ethernet2
(config-switch-if)# role delivery interface-name d1
(config-switch)# interface ethernet3
(config-switch-if)# role delivery interface-name d2
(config-switch)# interface ethernet4
(config-switch-if)# role delivery interface-name d3
```

5. Set a max **`overlap-policy-limit`** value, for example, ***2***.

```
(config-switch)# overlap-policy-limit 2
```

6. Create overlap policies using the same **`filter-interface`**, in this example, ***f1***.

```
(config-switch)# policy p1
(config-policy)# filter-interface f1
(config-policy)# delivery-interface d1
(config-policy)# action forward
(config-policy)# 1 match any

(config-switch)# policy p2
(config-policy)# filter-interface f1
(config-policy)# delivery-interface d2
(config-policy)# action forward
(config-policy)# 1 match any
```

7. Since the **overlap-policy-limit** value is ***2***, the third overlap policy will not allow the use of the same filter interface ***f1*** in the third policy, ***p3***. DMF throws a validation error.

```
(config-switch)# policy p3
(config-policy)# delivery-interface d3
(config-policy)# action forward
(config-policy)# filter-interface f1
Error: Validation failed: Filter interfaces used in more than 2 policies: f1
```

Show Commands

The following command example displays the configured policies listing two overlap policies **_p1_o_p2**.

```
(config-policy)# show policy
# Policy Name Action  Runtime Status               Type       Priority Overlap Priority Push VLAN Filter BW (truncated...)
-|-----------|-------|---------------------------|----------|--------|----------------|---------|--------- (truncated...)
1 _p1_o_p2    forward A component policy failed    Dynamic    100      1                3         10Gbps    (truncated...)
2 p1          forward all delivery interfaces down Configured 100      0                1         10Gbps    (truncated...)
3 p2          forward all delivery interfaces down Configured 100      0                2         10Gbps    (truncated...)
4 p3          forward inactive                     Configured 100      0                4         -         (truncated...)
```

To add filter interface ***f1*** to the third overlap policy, ***p3***, set the policy to **inactive**.

```
(config-switch)# policy p3
(config-policy)# inactive
(config-policy)# filter-interface f1
```

This results in an inactive policy ***p3*** being configured with filter interface ***f1***. Use the **show running-config policy** command to view the status.

```
(config-policy)# show running-config policy
! policy
policy p1
action forward
delivery-interface d1
filter-interface f1
1 match any

policy p2
action forward
delivery-interface d2
filter-interface f1
1 match any

policy p3
action forward
delivery-interface d3
filter-interface f1
inactive
1 match any
```

13.4.7 Exclude Expired Policies from Overlap Limit Calculation

If any two policies use the same filter interface and the same priority, then an additional dynamic policy is created to ensure the delivery of packets matching both of the original policies. There is a limit on how many overlap policies can be created and it is configurable with a range between `0` to `10` with a default value of `4`. Currently, the system excludes policies configured as ***inactive*** in the overlap policy limit calculation. With this new feature, the system excludes policies that have an expired duration from the overlap policies limit calculation.

CLI Configuration

No new configuration is necessary to use this feature. However, In order to achieve this improvement, the existing config validation `Filter interfaces used in more than n policies` was removed which is present in the policy creation of adding the filter interface after exceeding the max overlap limit number of policies.

Show Commands

Once the number of installed overlap policies reaches the maximum limit, any additional policies will not be installed. They are marked as inactive with the reason: `Filter interface fil1 cannot be used in more than 2 active policies`. In the example below, the overlap limit is `2`.

```
# show running-config overlap-policy-limit

! overlap-policy-limit
overlap-policy-limit 2
```

`policy1`, `policy2` and its overlap policy `_policy1_o_policy2` have been installed. Since the overlap policy limit has been reached, policy `policy3` is inactive.

```
# Policy Name        Action  Runtime Status Type       Priority Overlap Priority Push VLAN Filter BW Delivery BW
 Post Match Filter Traffic Delivery Traffic Services Installed Time          Installed Duration Ptp Timestamping
-|------------------|-------|--------------|----------|--------|----------------|---------|---------|---
--------|-------------------------|----------------|--------|----------------------|------------------|
----------------|
1 _policy1_o_policy2 forward installed      Dynamic    100      1                3         10Gbps    20Gbps
 -                       -                          2025-06-30 06:55:00 UTC 13 secs            False
2 policy1            forward installed      Configured 100      0                2         10Gbps    10Gbps
 -                       -                          2025-06-30 06:55:00 UTC 13 secs            False
3 policy2            forward installed      Configured 100      0                1         10Gbps    10Gbps
 -                       -                          2025-06-30 06:55:00 UTC 13 secs            False
4 policy3            forward inactive       Configured 100      0                4         10Gbps    10Gbps
 -                       -                                                                     False
```

The reason for policy `policy3` inactive is shown in both the `Config Status` and `Detailed Status`.

```
# show policy policy3
Policy Name                             : policy3
Config Status                           : Filter interface fil1 cannot be used in more than 2 active policies
Runtime Status                          : inactive
Detailed Status                         : Filter interface fil1 cannot be used in more than 2 active policies
Priority                                : 100
Overlap Priority                        : 0
# of switches with filter interfaces    : 1
# of switches with delivery interfaces : 1
# of switches with service interfaces   : 0
# of filter interfaces                  : 1
# of delivery interfaces                : 1
# of core interfaces                    : 0
# of services                           : 0
# of pre service interfaces             : 0
# of post service interfaces            : 0
Push VLAN                               : 4
Post Match Filter Traffic               : -
Total Delivery Rate                     : -
Total Pre Service Rate                  : -
Total Post Service Rate                 : -
Overlapping Policies                    : none
Component Policies                      : none
Timestamping enabled                    : False
~ Match Rules ~
# Rule
-|-----------|
1 1 match any

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF Switch  IF Name    State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time
-|------|-------|----------|-----|---|-------|-----|--------|--------|------------------|
1 fil1   filter1 ethernet12 up    rx  0       0     0        -

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s)  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# DMF IF Switch IF Name    State Dir Packets Bytes Pkt Rate Bit Rate Counter Reset Time
-|------|------|----------|-----|---|-------|-----|--------|--------|------------------|
1 del3   core1  ethernet14 up    tx  0       0     0        -

~ Service Interface(s) ~
None.

~ Core Interface(s) ~
None.

~ Failed Path(s) ~
None.
```

These inactive policies are displayed in the fabric errors as shown below with `Policy Name` and `Error`.

```
# show fabric errors policy-error
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Policy related error ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Policy Name Error
-|-----------|-------------------------------------------------------------------|
1 policy3     Filter interface fil1 cannot be used in more than 2 active policies
```

`policy2` is configured with an expiration duration of 180 seconds as shown below.

```
# show running-config policy policy2

! policy
policy policy2
  action forward
  delivery-interface del2
  filter-interface fil1
  start on-date-time 2025-06-30T06:54:00.014409+00:00 duration 180 delivery-count 0
  1 match any
```

After 180 seconds, `policy2` is marked as inactive due to its expired duration. Here policy `policy3` is installed automatically as the existing installed overlap policy `policy2` becomes expired.

```
# show policy
# Policy Name        Action  Runtime Status Type       Priority Overlap Priority Push VLAN Filter BW Delivery BW
 Post Match Filter Traffic Delivery Traffic Services Installed Time          Installed Duration Ptp Timestamping
-|------------------|-------|--------------|----------|--------|----------------|---------|---------|---
--------|-------------------------|----------------|--------|-----------------------|------------------|
----------------|
1 _policy1_o_policy3 forward installed      Dynamic    100      1                5         10Gbps    20Gbps
 -                       -                          2025-06-30 06:57:00 UTC 6 minutes, 4 secs  False
2 policy1            forward installed      Configured 100      0                2         10Gbps    10Gbps
 -                       -                          2025-06-30 06:57:00 UTC 6 minutes, 4 secs  False
3 policy2            forward inactive       Configured 100      0                1         10Gbps    10Gbps
 -                       -                                                                 False
4 policy3            forward installed      Configured 100      0                4         10Gbps    10Gbps
 -                       -                          2025-06-30 06:57:00 UTC 6 minutes, 4 secs  False
```

Considerations

This feature does not apply to policies that are expired because they exceeded the configured packet delivery count. Example below on how to use the delivery count on CLI.

```
(config-policy)# start now delivery-count 200
(config-policy)# show this

! policy
policy policy1
  action forward
  delivery-interface delivery1
  filter-interface filter1
  start on-date-time 2025-07-04T17:02:25.669872+00:00 duration 0 delivery-count 200
```

13.5 Viewing Information about Policies

Installing and activating overlapping policies may take more than a minute, depending on the number of overlapping policies and the number of rules in each policy.

13.5.1 Viewing Policy Flows

The **show policy-flow** command lists all the flows installed by the DANZ Monitoring Fabric (DMF) application on the switches in the monitoring fabric. The following is the command syntax:

```
show policy-flow policy_name
```

Flows are sorted on a per-policy basis. Each flow entry includes the configured policy name. The packet and byte count is affiliated with each flow entry, as shown in the following example:

```
controller-1# show policy-flow _P1_o_P2
# Policy Name Switch                                          Pkts Bytes Pri  T Match   Instructions
1 _P1_o_P2    DMF-CORE-SWITCH-1 (00:00:cc:37:ab:a0:90:71)     0    0     6401 1 in-port 16,vlan-vid 7 apply: name=_P1_o_P2 output: max-length=65535, port=15
2 _P1_o_P2    DMF-CORE-SWITCH-1 (00:00:cc:37:ab:a0:90:71)     0    0     6401 1 in-port 16,eth-type ipv6,vlan-vid 7 apply: name=_P1_o_P2 output: max-length=65
535, port=15
3 _P1_o_P2    DMF-DELIVERY-SWITCH-1 (00:00:cc:37:ab:60:d4:74) 0    0     6401 1 in-port 49,eth-type ipv6,vlan-vid 7 apply: name=_P1_o_P2 output: max-length=65
535, port=15,output: max-length=65535, port=1
4 _P1_o_P2    DMF-DELIVERY-SWITCH-1 (00:00:cc:37:ab:60:d4:74) 0    0     6401 1 in-port 49,vlan-vid 7 apply: name=_P1_o_P2 output: max-length=65535,
 port=15,output: max-length=65535, port=1
--------------------------------------------------------------------------------------output truncated-----------------------------------------------------
```

13.5.2 Viewing Packets Dropped by Policies

The **drops** option displays the current value of the transmit drop packet counters at the filter, delivery, and core interfaces for the specified policy, as shown in the following example:

```
controller-1# show policy p1 drops
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Filter Interface(s) Drops ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# IF  Switch IF               Name     state speed   Xmit Drops Pkt Count Xmit Drops Pkt Rate Rx Drops Pkt Count Rx Drops Pkt Rate
-|---|-----------------------|--------|-----|-------|--------------------|-------------------|------------------|-----------------|
1 f1  00:00:00:00:00:00:00:0c s12-eth1 up    10 Gbps 0                    0                   0                  0
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Delivery Interface(s) Drops ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# IF  Switch IF               Name     state speed   Xmit Drops Pkt Count Xmit Drops Pkt Rate Rx Drops Pkt Count Rx Drops Pkt Rate
-|---|-----------------------|--------|-----|-------|--------------------|-------------------|------------------|-----------------|
1 d1  00:00:00:00:00:00:00:0c s12-eth2 up    10 Gbps 0                    0                   0                  0
~ Core Interface(s) Drops ~
None.
~ Service Interface(s) Drops ~
None.
```

13.6 Using Rule Groups

DANZ Monitoring Fabric (DMF) supports using an IP address group in multiple policies, referring to the group by name in match rules. If no subnet mask is provided in the address group, it is assumed to be an exact match. For example, for an IPv4 address group, no mask is interpreted as a mask of ***/32***. For an IPv6 address group, no mask is interpreted as ***/128***.

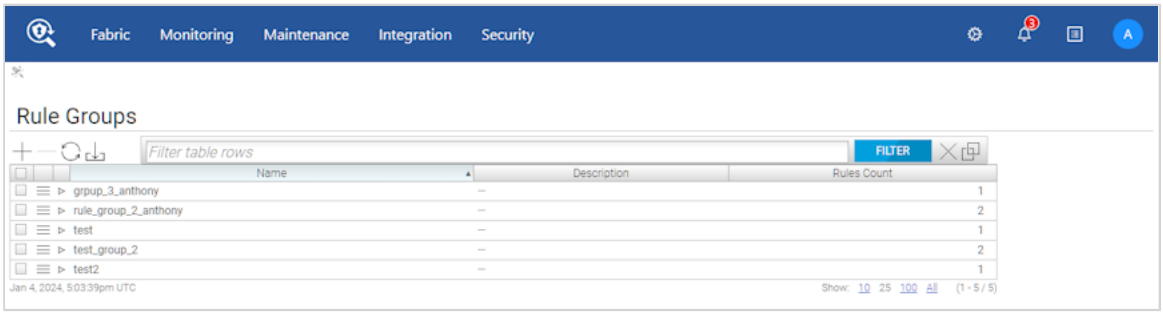
Identify only a single IP address group for a specific policy match rule. Address lists with both **src-ip** and **dst-ip** options cannot be used in the same match rule.

13.6.1 Using the GUI to Configure Rule Groups

To create an interface group from the **Monitoring > Interfaces** table, perform the following steps:

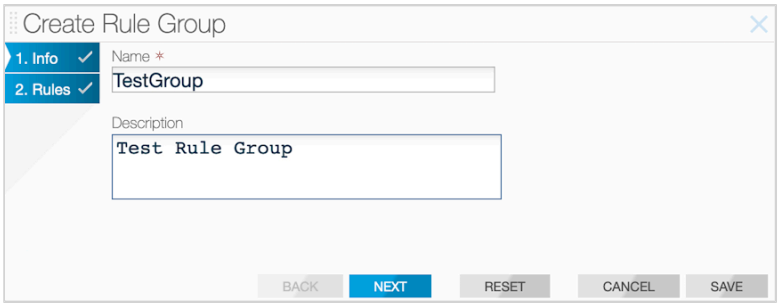
1. Select the **Monitoring > Rule Groups** option.

   Figure 13-17: Creating Rule Groups



2. On the **Rule Groups** table, select the **+** sign to create a new rule group.
3. In the pop-up menu, enter a preferred name for the rule group and, optionally, a description.

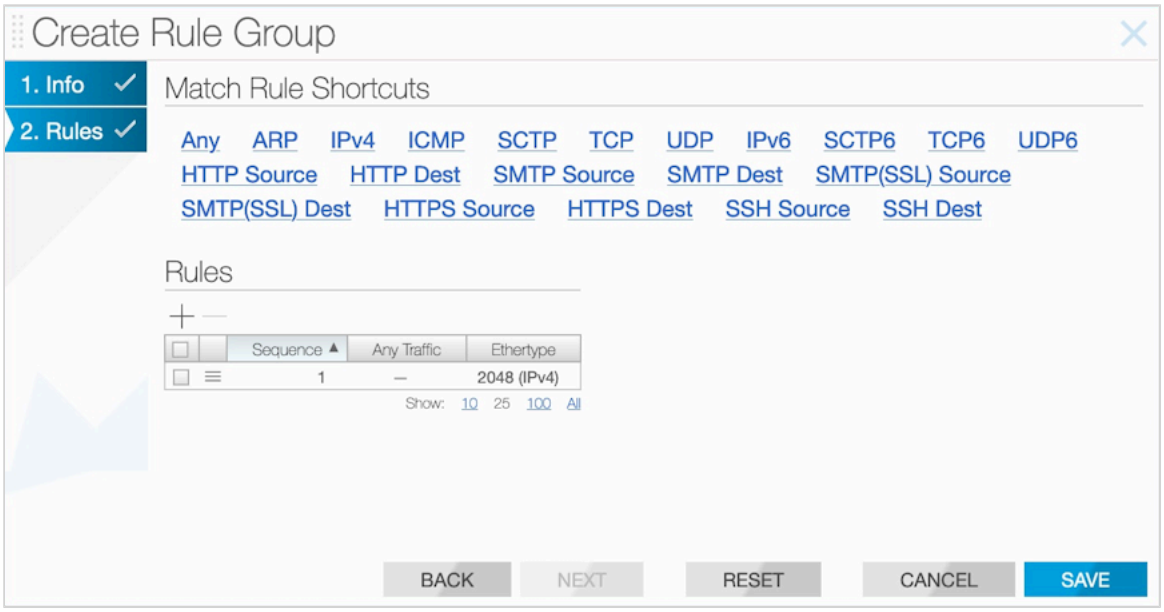
   Figure 13-18: Creating Rule Groups: Enter a Rule Group Name and Description



4. Select **NEXT** to add specific rules to the rule group.

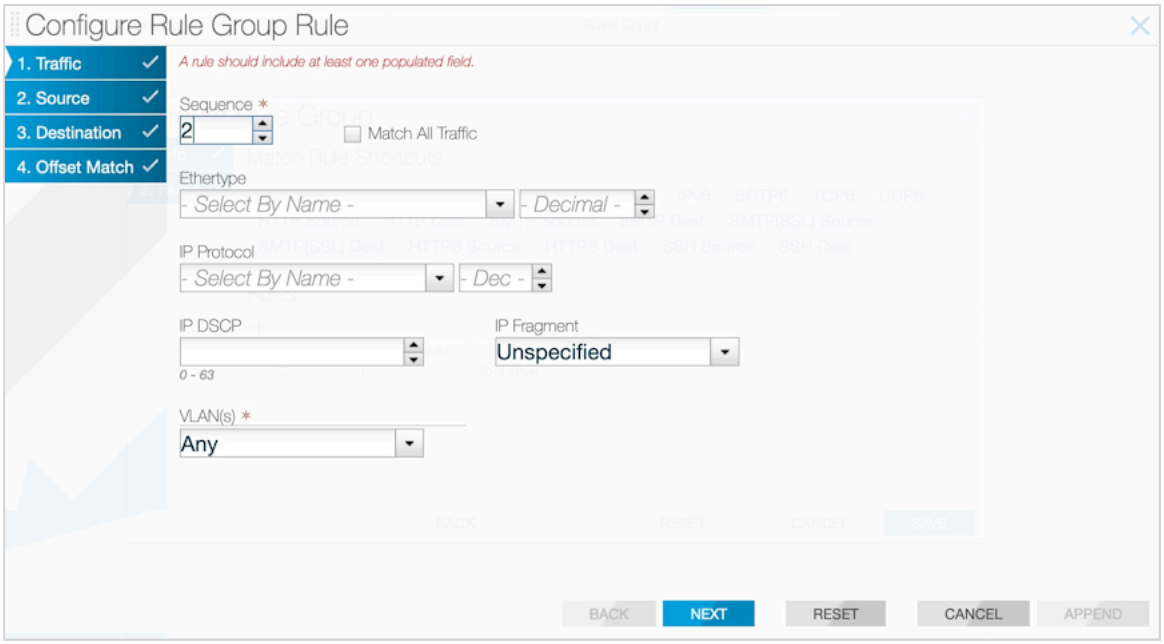
5. In this pop-up section, add predefined rules by selecting the options provided. In the example below, add a rule to match all IPv4 traffic by selecting **IPv4**.

Figure 13-19: Creating Rule Groups: Add a Predefined Rule to the Rule Group



6. As an alternative to the previous step, add custom rules by selecting the **+** sign under **Rules** and adding the necessary fields in the new pop-up screen.

Figure 13-20: Creating Rule Groups: Add Custom Rules to the Rule Group



7. Complete the dialog that appears to assign a descriptive name to the rule group.
8. Add this rule group to DANZ Monitoring Fabric (DMF) policies as a match condition.

13.6.2 Using the CLI to Configure Interface Groups

The following example describes configuring two interface groups: a filter interface group, **TAP-PORT-GRP**, and a delivery interface group, **TOOL-PORT-GRP**.

```
controller-1(config-switch)# filter-interface-group TAP-PORT-GRP
controller-1(config-filter-interface-group)# filter-interface TAP-PORT-1
controller-1(config-filter-interface-group)# filter-interface TAP-PORT-2
controller-1(config-switch)# delivery-interface-group TOOL-PORT-GRP
controller-1(config-delivery-interface-group)# delivery-interface TOOL-PORT-1
controller-1(config-delivery-interface-group)# delivery-interface TOOL-PORT-2
```

To view information about the interface groups in the DANZ Monitoring Fabric (DMF) fabric, enter the **show filter-interface-group** command, as in the following examples:

- **Filter Interface Groups**

```
controller-1(config-filter-interface-group)# show filter-interface-group
! show filter-interface-group TAP-PORT-GRP
# Name Big Tap IF Name                 Switch            IF Name    Direction Speed   State VLAN Tag
-|------------|-------------------------|-----------------|----------|---------|-------|-----|--------|
1 TAP-PORT-GRP TAP-PORT-1                DMF-CORE-SWITCH-1 ethernet17 rx        100Gbps up           0
2 TAP-PORT-GRP TAP-PORT-2                DMF-CORE-SWITCH-1 ethernet18 rx        100Gbps up           0
controller1(config-filter-interface-group)#
```

- **Delivery Interface Groups**

```
controller1(config-filter-interface-group)# show delivery-interface-group
! show delivery-interface-group DELIVERY-PORT-GRP
# Name Big Tap      IF Name         Switch                IF Name    Direction Speed  Ratelimit State Strip Forwarding Vlan
-|----------------|---------------|---------------------|----------|---------|------|---------|-----|---------------------|
1 TOOL-PORT-GRP     TOOL-PORT-1     DMF-DELIVERY-SWITCH-1 ethernet15 tx        10Gbps           up    True
2 TOOL-PORT-GRP     TOOL-PORT-2     DMF-DELIVERY-SWITCH-1 ethernet16 tx        10Gbps           up    True
controller-1(config-filter-interface-group)#
```

13.7 PTP Timestamping

DANZ Monitoring Fabric (DMF) rewrites the source MAC address of packets that match a policy with a 48-bit timestamp value sourced from a high-precision hardware clock.

- Connect a switch with a filter interface to a PTP network with a dedicated interface for the Precision Time Protocol. With a valid PTP interface, the switch will be configured in boundary clock mode and can sync the hardware clock with an available Grandmaster clock.

- Once configuring a policy to use timestamping, any packet matching on this policy will get its source MAC address rewritten with a timestamp value. The same holds true for any overlapping policy that carries traffic belonging to a user policy with timestamp enabled.

- The following options are available to configure a switch in boundary mode:
  - **domain**: Value for data plane PTP domain (0-255) (optional)
  - **Priority1**: Value of priority1 data plane PTP (0-255) (optional)
  - **Source IPv4 Address**: Used to restamp PTP messages from a switch to the endpoints (optional)
  - **Source IPv6 Address**: Used to restamp PTP messages from a switch to the endpoints (optional)

- The following options are available to configure an interface with role “ptp”:
  - **Announce Interval**: Set ptp announce interval between messages (-3,4). Default is 1 (optional)
  - **Delay Request Interval**: Set ptp delay request interval between messages (-7,8). The default is 5 (optional)
  - **Sync Message Interval**: Set ptp sync message interval between messages (-7,3). The default is 0 (optional)
  - **PTP Vlan**: VLANs used for Trunk or Access mode of operation for a ptp interface

- A policy should have enabled timestamping and have its filter interfaces on a switch with a valid PTP config to get its packets timestamped.

13.7.1 Platform Compatibility

DANZ Monitoring Fabric (DMF) supports the timestamping feature on 7280R3 switches.

Use the **`show switch all property`** command to check which switch in DMF fabric supports timestamping. If the following properties exist in the output, the feature is supported:

- `ptp-timestamp-cap-replace-smac`
- `ptp-timestamp-cap-header-48bit`
- `ptp-timestamp-cap-flow-based`

```
# show switch all property
# Switch                          ...   PTP Timestamp Supported Capabilities
-|----------------------------| ...  |-------------------------------------|
1 S1 (00:00:2c:dd:e9:96:2b:ff)  ...   ptp-timestamp-cap-replace-smac,
                                ...   ptp-timestamp-cap-header-64bit,
                                ...   ptp-timestamp-cap-header-48bit,
                                ...   ptp-timestamp-cap-flow-based,
                                ...   ptp-timestamp-cap-add-header-after-l2
2 S2 (00:00:cc:1a:a3:91:a7:6c)  ...
2 S3 (00:00:cc:1a:a3:c0:94:3e)  ...
```



Note: The CLI output example above is truncated for illustrative purposes. The actual output will differ.

13.7.2 Configuration

The following three sections describe the configuration for PTP and timestamping:

- PTP Switch Configuration
- PTP Interface Configuration
- Policy Configuration for Timestamping



Note: Configuring the PTP domain for the distribution of PTP-based time sync in the network is beyond the scope of this document. The assumption is the network exists with a PTP domain and all necessary components.

13.7.3 Configuring PTP Timestamping using the CLI

Configure the switch at a global level under the **config** submode in the CLI or for each switch under the **config-switch** submode. Irrespective of the place, it has the following options:

1. **Domain**: Set the data plane PTP domain. The default value is 0. Valid values are [0 to 255] inclusive.
2. **Priority1**: Set the value of priority1 data plane PTP. The default value is 128. Valid values are [0 to 255] inclusive.
3. **Source-ipv4-address**: This is the source IPv4 address used to restamp PTP messages from this switch to the endpoints. Some master clock devices do not accept default source IP (0.0.0.0). If so, configure it can to sync with such devices. The default is 0.0.0.0 .
4. **Source-ipv6-address**: This is the source IPv6 address used to restamp PTP messages from this switch to the endpoints. Some master clock devices do not accept default source IP (::/0). If so, configure it to sync with such devices. The default is ::/0 .

All fields are optional, and default values are selected if not configured by the user.

Global Configuration

The global configuration is a central place to provide a common switch config for PTP. It only takes effect after creating a ***ptp-interface*** for a switch. Under the **config** submode, provide PTP switch properties using the following commands:

```
> enable
# config
(config)# ptp priority1 0 domain 1 source-ipv4-address 1.1.1.1
```

Local Configuration

The local configuration provides a local PTP configuration or overrides a global PTP config for a selected switch. Select the switch using the command **switch *switch name***. PTP switch config (local or global) only takes effect after creating a ***ptp-interface*** for a switch. Under the **config-switch** submode, provide local PTP switch properties using the following commands:

```
(config)# switch eos
(config-switch)# ptp priority1 1 domain 2
```

13.7.4 Configuring PTP Timestamping using the GUI

Global Configuration

To view or edit the global PTP configuration, navigate to the DANZ Monitoring Fabric (DMF) Features page by selecting the **gear** icon.

Figure 13-21: DMF Menu Gear Icon



Scroll to the PTP Timestamping card and use **Edit** (**pencil** icon) to configure or modify the global PTP Timestamping settings.

Figure 13-22: DMF Features Page

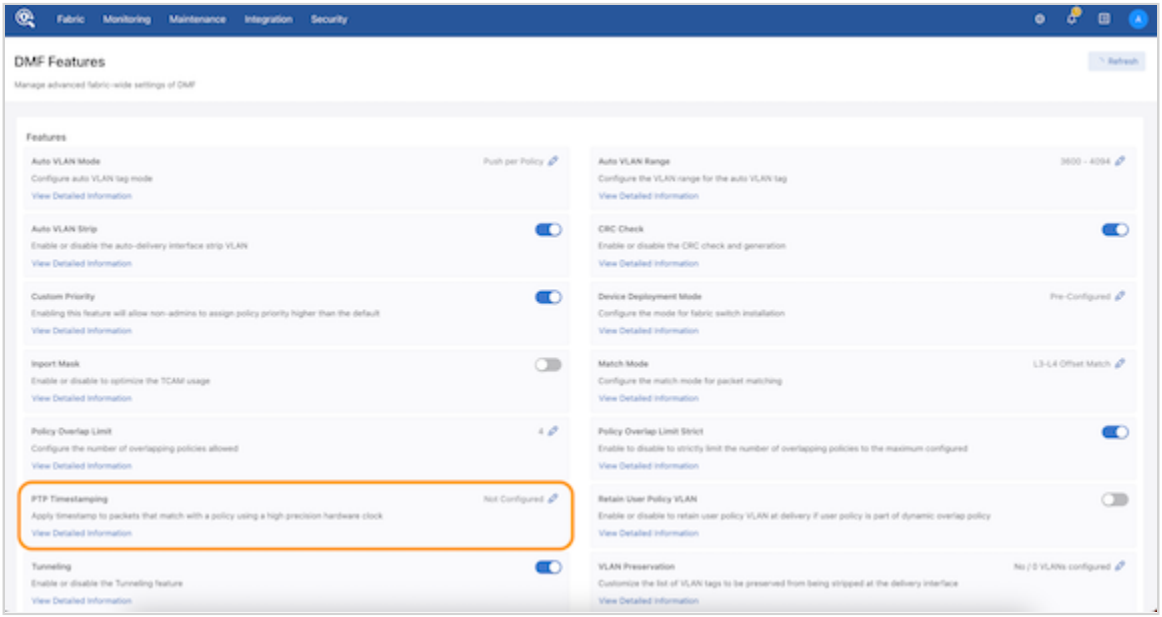
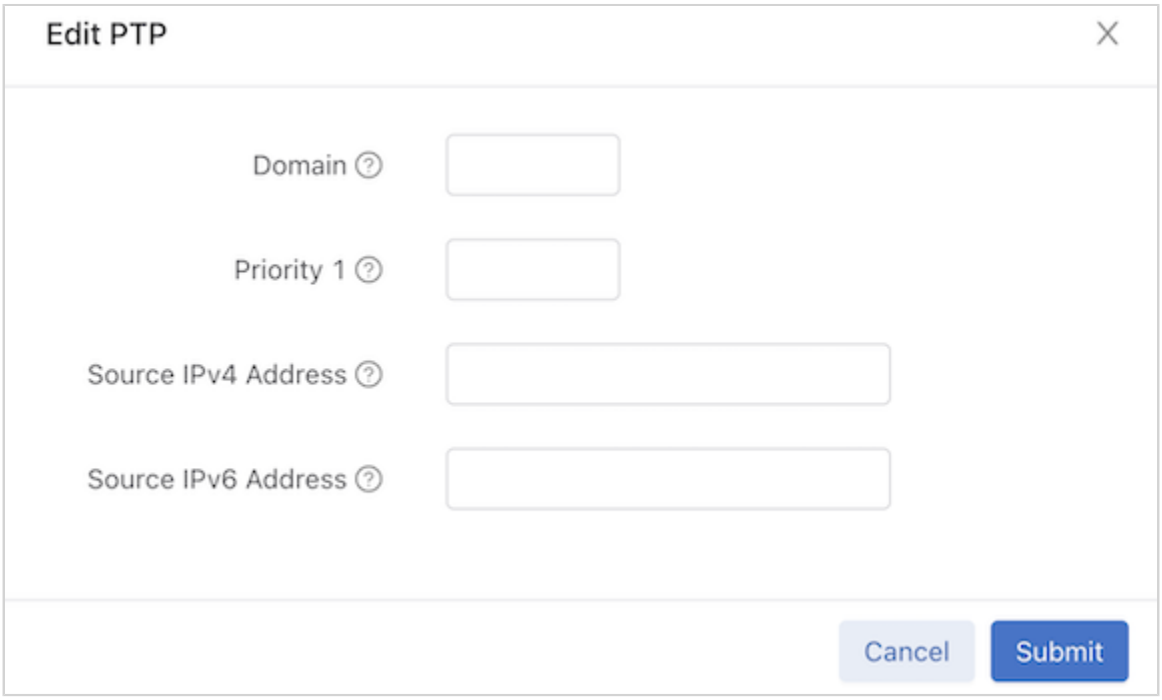


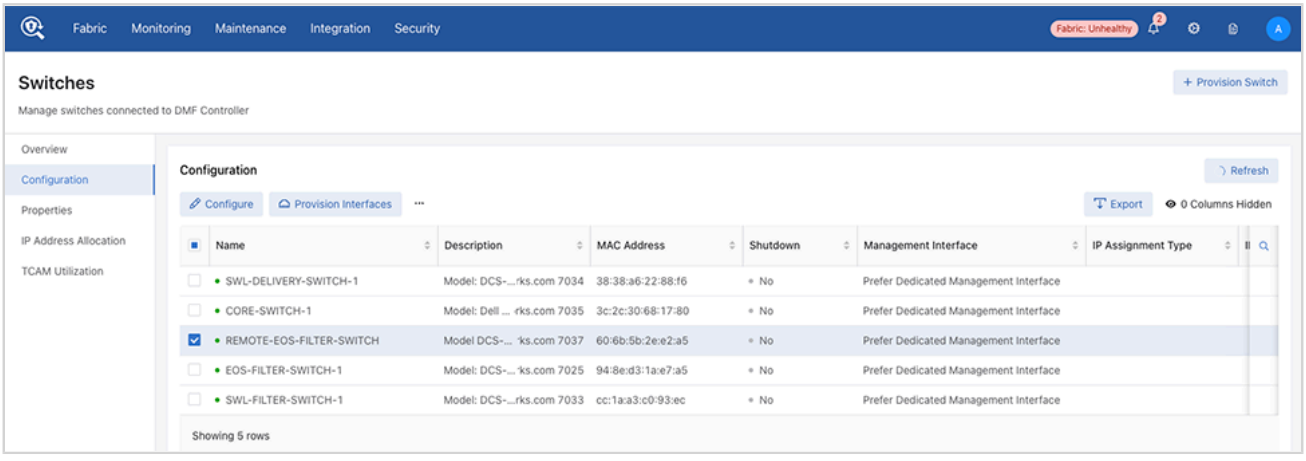
Figure 13-23: Edit PTP



Local Configuration

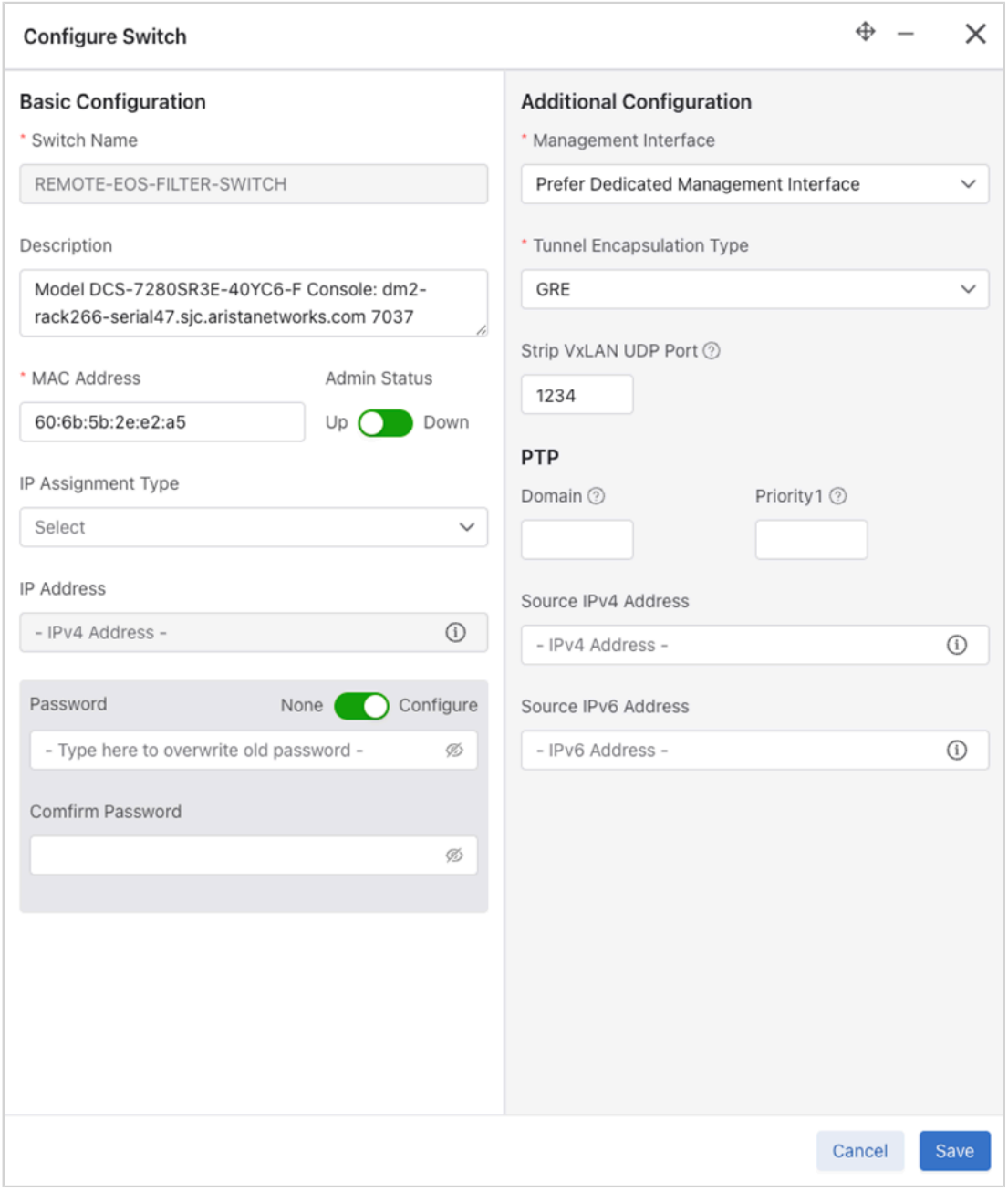
Provide a local PTP configuration for the switch or override the global PTP configuration for a selected switch while configuring or editing a switch configuration (under the PTP step) using **Fabric** > **Switch** > **Configuration**. Select a switch in the **Configuration** dashboard.

Figure 13-24: Configuration Dashboard



Select **Configure**.

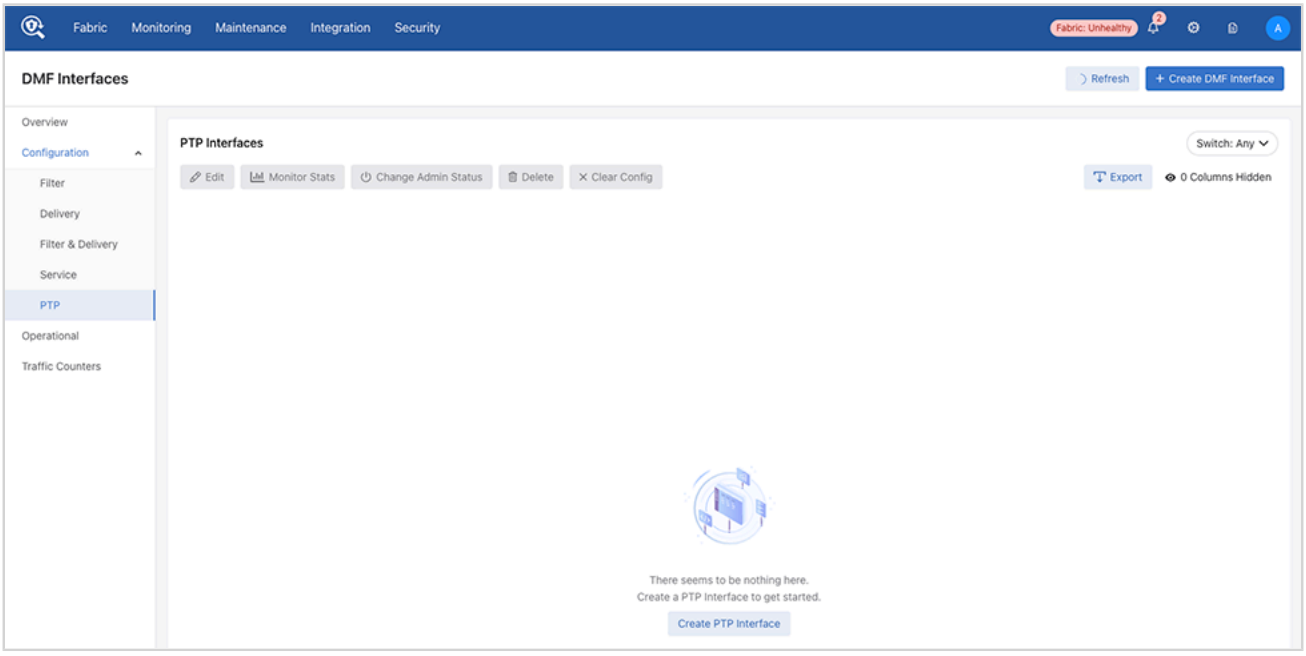
Figure 13-25: Configure Switch



PTP Interface Configuration

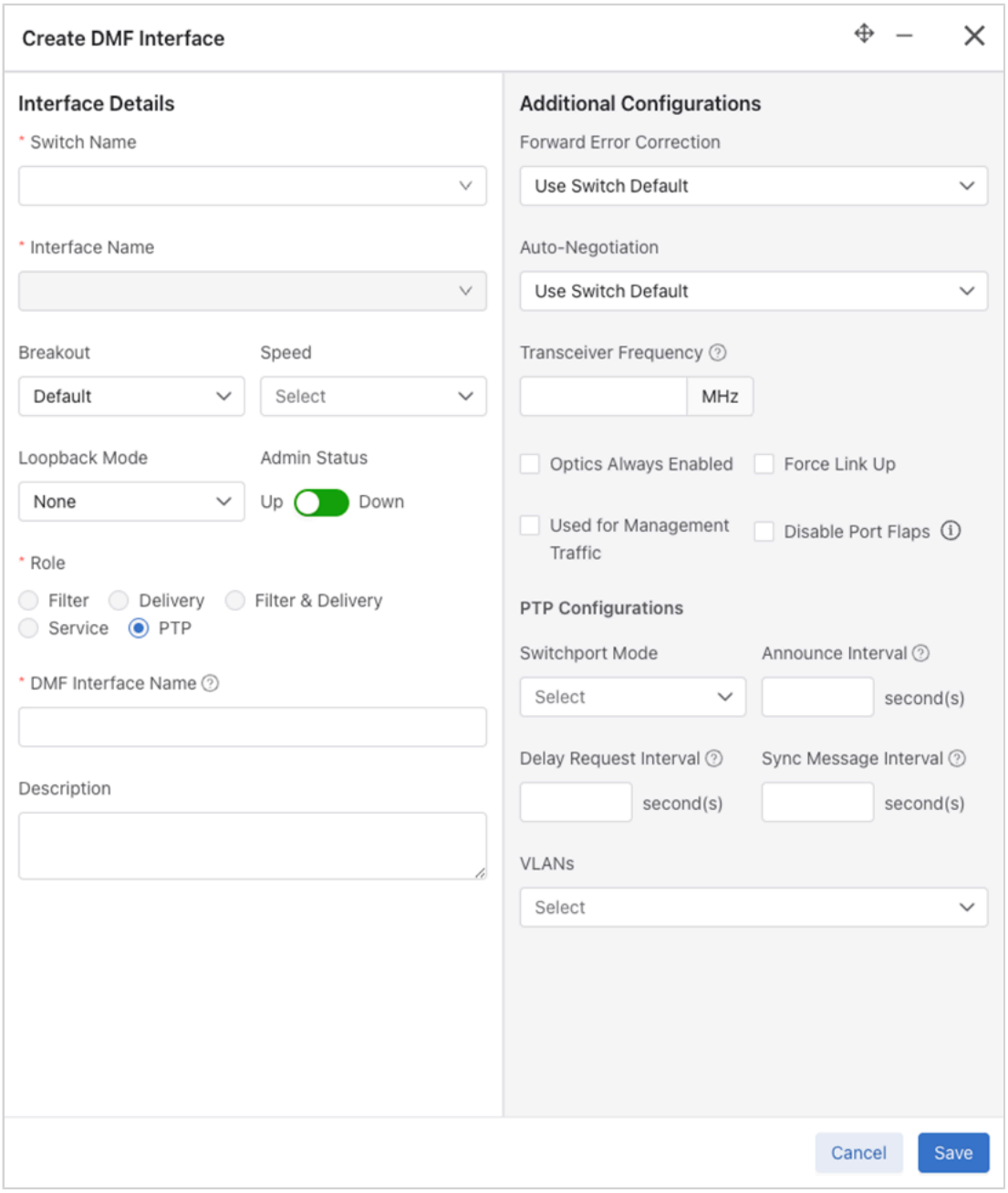
Configure a PTP Interface on the **Monitoring** > **Interfaces** > **Configuration** > **PTP** dashboard.

Figure 13-26: PTP Interfaces



Select **Create PTP Interface**.

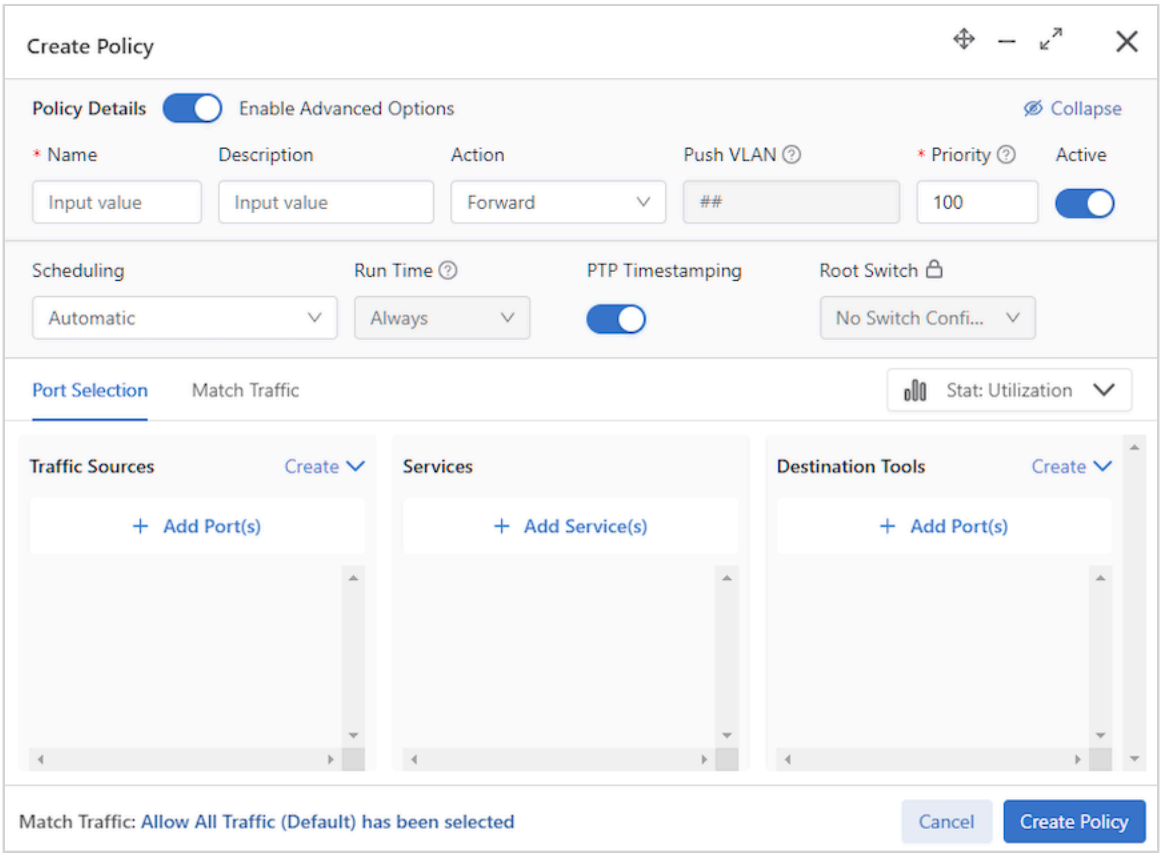
Figure 13-27: Create Interface



Timestamping Policy Configuration

DMF supports flow-based timestamping. This function requires programming a policy to match relevant traffic and enable timestamping for the matched traffic. In the **Create/Edit Policy** workflow (on the **Monitoring > Policies** page), use the **PTP Timestamping** toggle to enable or disable timestamping.

Figure 13-28: Create Timestamping Policy



13.7.5 PTP Interface Configuration

A switch that syncs its hardware clock using PTP requires a physical front panel interface configured as a PTP interface. This interface is solely responsible for communication with the master clock and has no other purpose.

To configure the PTP interface, select an interface on the switch, as illustrated in the following command.

```
(config-switch)#  interface Ethernet6/1
```

Use the **role** command to assign a ***ptp*** role and interface name and select ***switchport-mode*** for the specified interface.

```
(config-switch-if)# role ptp interface-name ptp1 access-mode announce-interval 1 delay-request-
interval 1 sync-message-interval 1
```

A switch port is required to configure a PTP interface. The options for switch port mode are:

- **trunk-mode**
- **access-mode**
- **routed-mode**

The **switchport mode** configuration for a PTP interface is necessary to match the PTP master switch's interface configuration. Configure the master switch to communicate PTP messages with or without a vlan tag. Use the trunk-mode with the appropriate ptp vlan when configuring the neighbor similarly. If the neighbor's interface is in switch-port access mode or routed mode, use either of these to match it on the filter switch.

Other fields are optional, using default values when no configuration is provided.

Optional fields:

- **announce-interval**: Set PTP to announce interval between messages [-3,4]. The default value is 1.
- **delay-request-interval**: Set PTP delay request interval between messages [-7,8]. The default value is 5.
- **sync-message-interval**: Set PTP sync message interval between messages (-7,3). The default value is 0.

Depending on the switch port mode selected for this interface, provide VLANs that will be associated with the selected ***ptp-interface*** using the following commands:

```
(config-switch-if)# ptp vlan 1
(config-switch-if)# ptp vlan 2
```

In routed switchport mode, we ignore the configured VLANs. In access switchport mode, the first VLAN is used for programming while ignoring the rest. In trunk switchport mode, all configured VLANs are programmed into the switch.

13.7.6 Policy Configuration for Timestamping

DANZ Monitoring Fabric (DMF) supports flow-based timestamping. This function requires programming a policy to match relevant traffic and enable timestamping for the matched traffic.

Create a policy using the command **`policy`** ***policy name***.

Under **`config-policy`** submode, enable timestamping using the following command:

```
(config-policy)# use-timestamping
```

13.7.7 L2GRE Encapsulation of Packets with Arista Timestamp Headers

L2GRE encapsulation of packets with Arista timestamp headers is an extension of an existing feature allowing DANZ Monitoring Fabric (DMF) to use intra-fabric L2GRE tunnels.

These tunnels enable forwarding unmodified production network packets over intermediate L3 networks used by DMF, which can now forward packets with Arista Networks Timestamp headers across L2GRE tunnels defined on EOS switches.

When a PTP header-based timestamping capable filter switch is in a remote location, and the remote filter switches connect to the centralized tool farm via L2GRE tunnels, timestamp the packets using PTP timestamping. These packets are encapsulated in an L2GRE header and sent to the core switch via the L2GRE tunnel. The timestamped packets will properly decapsulate at the remote end and be forwarded to the destination tools.



Note: DMF only supports PTP timestamping with L2GRE encapsulation with the header-based timestamping feature.

Please refer to the Tunneling Between Data Centers section on using L2GRE tunnels in DMF.

13.7.8 Using the CLI Show Commands

PTP State Show Commands

Use the **show switch** ***switch name*** **ptp** ***info*****|** ***masters*** **|** ***interface*** **|** ***local-clock*** command to obtain the PTP state of the selected switch.

The **show switch** ***switch name*** **ptp info** command summarizes the switch's PTP state and the PTP interfaces' status.

```
Controller# show switch eos ptp info
PTP Mode: Boundary Clock
PTP Profile: Default ( IEEE1588 )
Clock Identity: 0x2c:dd:e9:ff:ff:96:2b:ff
Grandmaster Clock Identity: 0x44:a8:42:ff:fe:34:fd:7e
Number of slave ports: 1
Number of master ports: 1
Slave port: Ethernet1
Offset From Master (nanoseconds): -128
Mean Path Delay (nanoseconds): 71
Steps Removed: 2
Skew (estimated local-to-master clock frequency ratio): 1.0000080070748882
Last Sync Time: 00:52:44 UTC Aug 09 2023
Current PTP System Time: 00:52:44 UTC Aug 09 2023
Interface       State        Transport    Delay
Mechanism
--------------- ------------ --------------- ---------
Et1 Slave        ipv4         e2e
Et47Master       ipv4         e2e
```

The **show switch** ***switch name*** **ptp master** command provides information about the PTP master and grandmaster clocks.

```
Controller# show switch eos ptp master
Parent Clock:
Parent Clock Identity: 0x28:99:3a:ff:ff:21:81:d3
Parent Port Number: 10
Parent IP Address: N/A
Parent Two Step Flag: True
Observed Parent Offset (log variance): N/A
Observed Parent Clock Phase Change Rate: N/A

Grandmaster Clock:
Grandmaster Clock Identity: 0x44:a8:42:ff:fe:34:fd:7e
Grandmaster Clock Quality:
Class: 127
Accuracy: 0xfe
OffsetScaledLogVariance: 0x7060
Priority1: 120
Priority2: 128
```

The **show switch *switch name* ptp interface *interface name*** command provides the PTP interface configuration and state on the device.

```
Controller# show switch eos ptp interface Ethernet1
Ethernet1
Interface Ethernet1
PTP: Enabled
Port state: Slave
Sync interval: 1.0 seconds
Announce interval: 2.0 seconds
Announce interval timeout multiplier: 3
Delay mechanism: end to end
Delay request message interval: 2.0 seconds
Transport mode: ipv4
Announce messages sent: 3
Announce messages received: 371
Sync messages sent: 4
Sync messages received: 739
Follow up messages sent: 3
Follow up messages received: 739
Delay request messages sent: 371
Delay request messages received: 0
Delay response messages sent: 0
Delay response messages received: 371
Peer delay request messages sent: 0
Peer delay request messages received: 0
Peer delay response messages sent: 0
Peer delay response messages received: 0
Peer delay response follow up messages sent: 0
Peer delay response follow up messages received: 0
Management messages sent: 0
Management messages received: 0
Signaling messages sent: 0
Signaling messages received: 0
```

The **show switch *switch name* ptp local-clock** command provides PTP local clock information.

```
Controller# show switch eos ptp local-clock
PTP Mode: Boundary Clock
Clock Identity: 0x2c:dd:e9:ff:ff:96:2b:ff
Clock Domain: 0
Number of PTP ports: 56
Priority1: 128
Priority2: 128
Clock Quality:
Class: 248
Accuracy: 0x30
OffsetScaledLogVariance: 0xffff
Offset From Master (nanoseconds): -146
Mean Path Delay: 83 nanoseconds
Steps Removed: 2
Skew: 1.0000081185368557
Last Sync Time: 01:01:41 UTC Aug 09 2023
Current PTP System Time: 01:01:41 UTC Aug 09 2023
```

Policy State Show Commands

Use the **`show policy`** command to view the timestamping status for a given policy.

```
> show policy
# Policy Name Action             Runtime Status Type       Priority Overlap Priority Push
 VLAN Filter BW Delivery BW Post Match Filter Traffic Delivery Traffic Services Installed Time
 Installed Duration Ptp Timestamping
-|-----------|------------------|--------------|----------|--------|----------------|------
---|---------|-----------|-------------------------|----------------|--------|-------------
-|------------------|----------------|
1 p1          unspecified action inactive       Configured 100      0               1         -
       -           -                         -
       True
```

13.7.9 Configuration Validation Messages

In **`push-per-policy mode`**, a validation exception occurs if a policy uses NetFlow managed-service with **records-per-interface** option and the same policy also uses timestamping. The following message appears:

```
Validation failed: Policy policy1 cannot have timestamping enabled along with header modifying
 netflow service.
Netflow service netflow1 is configured with records-per-interface in push-per-policy mode
```

In **`push-per-policy mode`**, a validation exception occurs if a policy uses the ipfix managed-service (using a template with **`records-per-dmf-interface key`**) and the same policy also uses timestamping. The following message appears:

```
Validation failed: Policy policy1 cannot have timestamping enabled along with header modifying
 ipfix service.
Ipfix service ipfix1 is configured with records-per-dmf-interface in push-per-policy mode
```

Only unicast source-ipv4-address or source-ipv6-address are allowed in the switch PTP config.

Examples of invalid ipv6 addresses: `ff02::1`, `ff02::1a`, `ff02::d`, `ff02::5`

```
Validation failed: Source IPv6 address must be a unicast address
```

Examples of invalid ipv4 addresses: `239.10.10.10`, `239.255.255.255`, `255.255.255.255`

```
Validation failed: Source IPv4 address must be a unicast address
```

13.7.10 Troubleshooting

A policy programmed to use timestamping can fail for the following reasons:

1. The filter switch does not support syncing its hardware clock using PTP.
2. An unconfigured PTP interface or the interface is inactive.
3. The PTP switch configuration or PTP interface configuration is invalid or incomplete.
4. Configuring the PTP interface on a logical port (Lag or Tunnel).

Reasons for failure will be available in the runtime state of the policy and viewed using the **`show policy`** ***`policy name`*** command.

As the Platform Compatibility Section describes, use the **`show switch all properties`** command to confirm a switch supports the feature.

13.7.11 Limitations

The source MAC address of the user packet is re-written with a 48-bit timestamp value on the filter switch.

This action can exhibit the following behavior changes or limitations:

1. Dedup managed service will not work as expected. A high-precision timestamp can be different for duplicate packet matching on two different filter interfaces. Thus, the dedup managed service will consider this duplicate packet to be different in the L2 header. To circumvent this limitation, use an anchor/offset in the dedup managed-service config to ignore the source MAC address.
2. Any Decap managed service except for **`decap-l3-mpls`** will remove the timestamp information header.
3. The user source MAC address is lost and unrecoverable when using this feature.
4. The **`rewrite-dst-mac`** feature cannot be used on the filter interface that is part of the policy using the timestamping feature.
5. In push-per-filter mode, if a user has src-mac match condition as part of their policy config, the traffic will not be forwarded as expected and can get dropped at the core switch.
6. The **`in-port masking`** feature will be disabled for a policy using PTP timestamping.
7. Logical ports (Lag/Tunnel) as PTP interfaces are not allowed.

Appendix A

Stenographer Reference for DMF Recorder Node

This appendix provides information about composing Stenographer queries and submitting them through REST API.

A.1 Stenographer Query Syntax

The DANZ Monitoring Fabric (DMF) Recorder Node accepts Stenographer queries using a syntax based on the Berkeley Packet Filter (BPF) syntax. When entering a malformed BPF string, the recorder node will respond with an error. The entire BPF grammar is not supported, but query strings can be composed using the predicates in the following table.

Table 14: Table 1: Supported Stenographer BPF Query Strings

BPF Predicate	Value	Description
before ***value***	time	string before the specified time
before ***value*** m ago	duration	before *value* minutes ago
before ***value*** h ago	duration	before *value* hours ago
before ***value*** d ago	duration	before *value* days ago
before ***value*** w ago	duration	before *value* weeks ago
after ***value***	time string	after the specified time
after ***value*** m ago	duration	after *value* minutes ago
after ***value*** h ago	duration	after *value* hours ago
vlan ***value***	VLAN ID	match the specified VLAN tag (outer, inner, or inner inner)
outer vlan ***value***	VLAN ID	match the specified outer VLAN tag
inner vlan ***value***	VLAN ID	match the specified inner VLAN tag (or middle tag of triple-tagged packets)
inner vlan ***value***	VLAN ID	match the specified innermost VLAN tag of triple-tagged packets
src mac ***value***	MAC address	match the specified MAC address in typical colon-delimited form (e.g. 11:22:33:44:55)
dst mac ***value***	MAC address	match the specified MAC address in typical colon-delimited form (e.g. 11:22:33:44:55)
mpls ***value***	MPLS label	match the specified MPLS label
src host ***value***	IPv4/v6 address	match the specified source address exactly
dst host ***value***	IPv4/v6 address	match the specified destination address exactly
src net ***value***	IPv4/v6 address	match the specified source address with an optional CIDR mask. All octets of address must be specified, e.g. good → 1.2.3.0/24, bad → 1.2.3/24
src net ***value*** mask ***value***	IPv4/v6 address	match the specified source address with masked with the specified address
dst net ***value***	IPv4/v6 address	match the specified destination address with an optional CIDR mask. All octets of address must be specified, e.g. good → 1.2.3.0/24, bad → 1.2.3/24
dst net value mask ***value***	IPv4/v6 address	match the specified destination address with masked with the specified address
ip proto ***value***	protocol number	match the specified IP protocol number
icmp		match ICMP packets (shortcut for “ip proto 1”)
tcp		match TCP packets (shortcut for “ip proto 6”)

BPF Predicate	Value	Description
udp		match UDP packets (shortcut for “ip proto 17”)
src port ***value***	transport port number	match the specified transport port number
dst port ***value***	transport port number	match the specified transport port number
cid ***value***	Community ID	match the provided community ID in standard version:base-64 encoded form (e.g. 1:hO+sN4H +MG5MY/8hIrXPqc4ZQz0=)
policy ***value***	DMF policy name	match the forwarding VLAN(s) of the specified DMF policy. Only supported through the DMF Controller. Not supported when using a Recorder Node REST API directly.
filter-interface ***value***	DMF filter interface name	match the forwarding VLAN of the specified filter interface. Only supported through the DMF Controller. Not supported when using a Recorder Node REST API directly.
event ***value***	Recorder Node event name	match the time range of the specified event. Only supported through the DMF Controller. Not sup- ported when using a Recorder Node REST API directly.
and		logical “and”
&&		logical “and”
or		logical “or”
||		logical “or”
(		begin grouping
)		end grouping

A.2 Example Stenographer Queries



Note: Arista Networks recommends always using a specific time range in each query.

After two hours ago but before one hour ago, search for all packets to or from Google DNS (8.8.8.8).

```
(after 2h ago and before 1h ago) and (src host 8.8.8.8 or dst host 8.8.8.8)
```

In the last twenty-four hours, search for all SSH (TCP port 22) packets destined for IP 10.4.100.200.



Note: This will not match any SSH packets from 10.4.100.200.

```
after 24h ago and dst host 10.4.100.200 and tcp and src port 22
```

Within the last five minutes, search for all packets to or from 10.1.1.100. And, in the five minutes before that, search for all packets to or from 10.1.100.101.

```
(after 5m ago and (src host 10.1.1.100 or dst host 10.1.1.100)) or (after 10m ago and before 5m
 ago
and (src host 10.1.1.101 or dst host 10.1.1.101))
```

Within the timespan of event `abc` and within the last hour, search for all SSH (TCP port 22) packets destined for IP 1.2.3.4.

```
(event abc or after 1h ago) and dst host 1.2.3.4 and tcp and dst port 22
```

Within the timespan defined by the intersection of events `abc` and `def`, search for all packets sent from any IP in subnet 1.2.3.0/24 seen on filter interface `xyz`.

```
(event abc and event def) and filter-interface xyz and src net 1.2.3.0/24
.. note::
To use the filter-interface predicate, the DMF Controller must be in the push-per-filter Auto
VLAN mode.
```

Within the last five minutes, search for all packets sent from IP 1.2.3.4 to the DANZ Monitoring Fabric (DMF) Recorder Node using DMF policy `abc`.

```
after 5m ago and policy abc and src host 1.2.3.4
.. note::
To use the policy predicate the DMF Controller must be in the push-per-policy or push-per-
filter Auto VLAN mode. When in push-per-policy auto-vlan-mode, the policy's forwarding tag will
be queried. When in push-per-filter mode, the forwarding tags of the filter interfaces used in
the policy are queried.
```

Within the last five minutes, search for all packets with any VLAN tag 100.

```
after 5m ago and vlan 100
```

Within the last five minutes, search for all packets with an outer VLAN tag 100.

```
after 5m ago and outer vlan 100
```

Within the last five minutes, search for all packets with an inner (or middle) VLAN tag 100.

```
after 5m ago and inner vlan 100
```

Within the last five minutes, search for all triple-tagged packets with innermost VLAN tag 100.

```
after 5m ago and inner inner vlan 100
```

Within the last five minutes, search for packets belonging to a flow with community ID of 1:hO+sN4H +MG5MY/8hIrXPqc4ZQz0=.

```
after 5m ago and cid 1:hO+sN4H+MG5MY/8hIrXPqc4ZQz0=
```

This matches packets in each direction of the flow, if applicable.

Within the last five minutes, search for all L2 broadcast packets originating from MAC address 11:22:33:44:55:66.

```
after 5m ago and src mac 11:22:33:44:55:66 and dst mac ff:ff:ff:ff:ff:ff
```

Appendix B

DMF Recorder Node REST API

The REST server is available over HTTPS on the default port (***443***) using either of the two authentication methods supported:

- HTTP basic: The client presents a valid username and password for the Controller connected with the Recorder. The DANZ Monitoring Fabric (DMF) Recorder Node verifies at the DMF Controller if the provided username and password are valid and have sufficient privileges to use the Recorder REST API.
- Authentication tokens: The DMF Recorder Node REST API accepts revocable authentication tokens as an alternative to HTTP basic. Valid authentication tokens are configured in the Controller and pushed down to the DMF Recorder Node using a gentable. Any client with a valid authentication token can query the DMF Recorder Node REST API without real-time consultation with the Controller.

Some APIs accept a Stenographer query string as input or return a Stenographer query string as output. A Stenographer query string is a BPF-like syntax for defining the scope of a query. Packets that match this scope are included in the query result or operation. For details about the Stenographer query syntax supported by the DMF Recorder Node, refer to the Stenographer Reference for DMF Recorder Node. The DMF Recorder Node provides a REST API so clients can look up packets and metadata. The REST server runs securely (HTTPS) on TCP port ***443***.

B.1 Authentication

Clients must either authenticate using a valid DANZ Monitoring Fabric (DMF) Controller username and password over HTTP Basic or with an authentication token configured on the DMF controller specifically for DMF Recorder Node REST API authentication.

B.1.1 Basic HTTP Authentication

Use a valid DANZ Monitoring Fabric (DMF) Controller username and password to authenticate with a DMF Recorder Node over its REST API. The recorder node delegates authentication to the DMF Controller. If the username and password provided are valid, the recorder node authorizes the user for the invoked recorder node REST endpoint.

In the following example, the query uses the HTTP Basic method of authentication:

```
$ curl https://1.2.3.4/query/window -u admin:12345 -k | python -m json.tool
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 79 100 79 0 0 143 0 --:--:-- --:--:-- --:--:-- 143
{
"begin": "2019-01-23 15:15:23 +0000 UTC",
"end": "2019-02-04 17:39:52 +0000 UTC"
}
```

In this example, the recorder node IP address is ***1.2.3.4***. The username on the DMF Controller is **`admin`**, and the password is **`12345`**.

B.1.2 Authentication with an Authentication Token

An authentication token is primarily designed for third-party applications and automation scripts where creating an account or storing a username and password is undesirable. This method can also allow a DANZ Monitoring Fabric (DMF) Recorder Node access if the management network connection to the Controller is disrupted.

To create an authentication token, log in to the DMF Controller associated with the recorder node, then perform the following steps:

1. Change to config mode on the active DMF Controller.

```
controller-1# configure
controller-1(config)#
```

2. Define the authentication token using a unique name.

```
controller-1(config)# recorder-node auth token my-token
Auth : my-token
Token : the-secret-token
```



Note: This name does not need to be secret. This example uses the name my-token.

The Controller generates a secret token (in this example, the-secret-token). Treat this token as private. Anyone who presents it to the DMF Recorder Node can use the DMF Recorder Node REST APIs.



Note: Only the non-reversible hash of this token is stored on the DMF Recorder Node and Controller. There is no way to recover the token if it is lost. (See below for how to revoke the token if it is lost or compromised.)

The Controller stores the token hash and the name assigned, viewable by entering the **show running-config recorder-node** command, as in the following example:

```
controller-1(config)# show running-config recorder-node auth token
! recorder-node
recorder-node auth token my-token $2a$12$pXm62tl5rMD8c4vSrzU6X.DTjeoBmRUwZvTkvNXatsZ8TFb4PxanC
```

If the token is lost or compromised, remove it from the Controller, and the Controller will fail any attempt to authenticate to the recorder using the token.

```
controller-1(config)# no recorder-node auth token my-token
controller-1(config)# show running-config recorder-node auth token
controller-1(config)#
```

The following example illustrates a query using the authentication token method. The authentication token is defined in the HTTP request as the value of the cookie header.

```
$ curl https://1.2.3.4/query/inventory/window --header "Cookie:plaintext-secret-auth-token" -k |
python
-m json.tool
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 79 100 79 0 0 83 0 --:--:-- --:--:-- --:--:-- 83
{
"begin": "2019-01-23 15:15:23 +0000 UTC",
"end": "2019-02-04 17:50:46 +0000 UTC"
}
```

In this example, the DMF Recorder Node IP address is ***1.2.3.4***. The authentication token has already been generated on the DMF controller associated with the recorder node and is included in the cookie header as **`plaintext-secret-auth-token`**.

Only include the plaintext authentication token, not the token hash, saved in the controller running configuration. If the plaintext token is unknown, revoke access for the token and generate a new one. Note the plaintext value displayed after generating the new token.

B.2 DMF Recorder Node API Headers

The supported REST API HTTP header entries are listed in the following table.

Table 15: DANZ Monitoring Fabric (DMF) Recorder Node REST API HTTP Headers

Header	Value Type	Description
Steno-Limit-Bytes:value	integer	max number of bytes to accept in a packet query response
Steno-Limit-Packets:value	integer	max number of packets to accept in a packet query response
Cookie:value	string	auth token to use in lieu of HTTP basic auth

B.3 DMF Recorder Node REST APIs

The supported DANZ Monitoring Fabric (DMF) Recorder Node REST APIs are listed below.

B.3.1 Ready

`/ready`

- Description: Is the DANZ Monitoring Fabric (DMF) Recorder Node able to accept queries? Return payload indicates progress towards start up completion.
- HTTP Method: GET
- Request Payload:
- Return MIME Type:
- Return Payload:

```
{
"current-value": <int>,
"max-value": <int>,
"percent-complete": <float>
}
```

- Return Status Code:
  - 200, ready
  - 503, not ready

B.3.2 Query Window

`/query/window`

- Description: Get timestamp of oldest and newest packet available for query.
- HTTP Method: GET
- Request Payload:
- Return MIME Type: application/json
- Return Payload:

```
{
"begin" : <RFC-3339>,
"end" : <RFC-3339>
}
```

- Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.3.3 Query Size

`/query/size`

- Description: Get count and aggregate size of packets matching provided filter.
- HTTP Method: POST
- Request Payload: Stenographer query string
- Return MIME Type: application/json
- Return Payload:

```
{
"packet-count" : <int>,
"aggregate-size" : <int>
}
```

- Return Status Code:
  - 200, success 400, input error
  - 500, internal error
  - 503, not ready

B.3.4 Query Application

`/query/application`

- Description: Perform DPI on packets matching provided filter. DPI is performed using nDPI.
- HTTP Method: POST
- Request Payload: Stenographer query string
- Return MIME Type: application/json
- Return Payload: Defined by nDPI
- Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.3.5 Query Packet

`/query/packet`

- Description: Download pcap of packets matching provided filter.
- HTTP Method: POST
- Request Payload: Stenographer query string
- Return MIME Type: application/vnd.tcpdump.pcap
- Return Payload: .pcap file
- Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.3.6 Query Analysis Filter Stenographer Analysis Type

`/query/analysis[filter="<stenographer-query-string>"][type ="<analysis-type>"]`

- Description: Perform an analysis on the packets matching the stenographer query string. Supported values for *analysis-type* are:
  - analysis_http_tree
  - analysis_http_stat
  - analysis_http_req_tree
  - analysis_http_srv_tree
  - analysis_dns_tree
  - analysis_hosts
  - analysis_conv_ipv4
  - analysis_conv_ipv6
  - analysis_conv_tcp
  - analysis_conv_udp
  - analysis_rtp_streams
  - analysis_sip_stat
  - analysis_conv_sip
  - analysis_tcp_packets
  - analysis_tcp_flow_health
- HTTP Method: GET
- Request Payload:
- Return MIME Type: application/json
- Return Payload: Determined by the analysis type selected.
- Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.3.7 Query Replay Request Stenographer Real-time Boolean

/query/replay/request[filter="<stenographer-query-string>"][real-time="<boolean>"]

- Description: Asynchronously request packets matching filter be replayed into the monitoring fabric. Replay is performed using tcp replay.
- HTTP Method: POST
- Request Payload:
- Return MIME Type: application/json
- Return Payload:

```
{
"id" : <int>,
"message": <string>
}
```

- Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.3.8 Query Replay Request Filter Stenographer Integer

/query/replay/request[filter="<stenographer-query-string>"][speed-mbps="<int>"]

- Description: Asynchronously request packets matching filter be replayed into the monitoring fabric. Replay is performed using tcp replay.
- HTTP Method: POST
- Request Payload:
- Return MIME Type: application/json
- Return Payload:

```
{
"id" : <int>,
"message": <string>
}
```

- Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.3.9 Query Replay Done

`/query/replay/done`

- Description: Check the status of a replay matching the provided ID. Message contains replay result from tcp replay.
- HTTP Method: POST
- Request Payload: Replay ID
- Return MIME Type: application/json
- Return Payload:

```
{
"id" : <int>,
"done" : <boolean>,
"message": <string>
}
```

- Return Status Code:
  - 200, success
  - 400, input error
  - 404, replay ID unknown
  - 406, replay not done
  - 500, internal error
  - 503, not ready

B.3.10 Erase Packet Filter Stenographer Query String

`/erase/packet[filter="<stenographer-query-string>"]`

- Description: Erase packets matching the provided filter. Note that any packet not matching the filter but in the same packet file of a packet matching the filter will also be deleted.
- HTTP Method: POST
- Request Payload:
- Return MIME Type: application/json
- Return Payload:

```
{
"bytes-erased" : <int>,
"message" : <string>
}
```

- Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.3.11 Event Update Trigger Boolean Name Integer

/event/update[trigger="<boolean>"][name="<string>"][pre-buffer-minutes="<int>"]

- Description: Trigger or terminate the named event. Set pre-buffer-minutes to 0 to use the available pre-buffer.
- HTTP Method: POST
- Request Payload:
- Return MIME Type: application/json
- Return Payload:

```
{
"message" : <string>,
"event-queued" : <boolean>
}
```

- Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.3.12 Abort Query

`/abort/query`

- Description: Terminate a particular query defined by the provided Stenographer query string.
- HTTP Method: POST
- Request Payload:
- Return MIME Type: application/json
- Return Payload:

```
{
"message" : <string>
}
```

- Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.3.13 Abort All Query

`/abort-all/query`

- Description: Terminate all running queries.
- HTTP Method: POST
- Request Payload:
- Return MIME Type: application/json
- Return Payload:

```
{
"message" : <string>
}
```

- Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.3.14 Queries

`/queries`

- Description: Determine the currently running queries, enumerated by the Stenographer query string of the query.
- HTTP Method: GET
- Request Payload:
- Return MIME Type: application/json
- Return Payload:

```
{
"queries" : [
<stenographer-query-string>, ...
]
}
```

- Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.3.15 Status Query

`/status/query`

- Description: Determine how far a given query has progressed. This can be used to estimate the time remaining to run the query.
- HTTP Method: GET
- Request Payload:
- Return MIME Type: application/json
- Return Payload:

```
{
"query" : <stenographer-query-string>, "current-value" : <int>,
"max-value" : <int>, "percent-complete" : <float>
}
```

- Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.3.16 Status All

`/status/all`

- Description: Determine how far all queries have progressed. This can be used to estimate the time remaining to run the queries.
- HTTP Method: GET
- Request Payload:
- Return MIME Type: application/json
- Return Payload:

```
{
"queries" : [
{
"query" : <stenographer-query-string>,
"current-value" : <int>,
"max-value" : <int>,
"percent-complete" : <float>
},
...
]
}
```

- • Return Status Code:
  - 200, success
  - 400, input error
  - 500, internal error
  - 503, not ready

B.4 Recorder Node Recording State API

The Recorder Node (RN) recording statistics API on the DANZ Monitoring Fabric (DMF) Controller includes information about the operational state of the ongoing recording.

The information is available in **`device/state/packet-recorder/recording/recording-state`** where **`recording-state`** is the container holding the opstate information.



Note: This is a simplified version of the path; the path requires the **`name`** and **`type`** to be provided at the **`device`** to select a specific RN device to display state information.

Within the container, there are the following two values:

- **`state`**: an enumeration describing the recording opstate that has one of the following values: **`initializing`**, **`ready`**, **`active`**, and **`stopped`**.
- **`description`**: a string describing the state in more detail.

The following table lists the possible recording state values and describes how to interpret each scenario.

state	**description**	**Interpretation**
`initializing`	`Recording application syncing files, {}% complete.`	The application is synchronizing previously recorded packets, so it cannot record yet. The formatted value is the percentage of synchronization completed so far.
`initializing`	`Recording application is starting up.`	The application has finished synchronization and is now completing the setup for recording.
`ready`	`Ready to record, no recordable traffic received.`	No traffic has been received after the application completed initializing.
`active`	`Recording traffic.`	Traffic is being received and recorded to disk.
`stopped`	`Recording threads not running.`	Recording threads were interrupted or terminated, so recording has stopped.
`stopped`	`Recording not enabled.`	Recording is not enabled, and packets are not recorded.
`stopped`	`Disk space exhausted with stop-and-wait mode configured. Awaiting user input.`	Recording has stopped as disk space has been exhausted, and the removal policy has been configured as stop-and-wait. Delete packets to resume recording.
`stopped`	`Recording state not found.`	The recording application isn't running, so there is no recording state information.

When the state is not active, an RN warning is generated on the DMF Controller to indicate that a nonactive RN is connected to the fabric.



Note: Any other state than `active` is considered to be inactive. This is not necessarily indicative of a problem. However, you should take action accordingly using the provided interpretations.

Additionally, the type of logical disk volume used by each recording thread is part of the RN recording statistics API on the DMF Controller.

This information is available in **device/state/packet-recorder/recording/recording-thread/disk** where a **disk** is an enumeration having one of two values, **primary** or **backup**, if the thread is recording to the primary or backup configured logical disk volume, respectively. This is a simplified version of the path. The disk type does not imply whether the disk volume is local or remote, as the storage configuration determines this.

Show Commands

The **show recorder-node device *rn name* recording-health** command includes the state from the recording state and the disk type for each recording thread. Use the **details** option of this command to view the description of the recording state in addition to the previously stated information.

The following examples highlight the additional information in **bold** for illustrative purposes.

```
C1> show recorder-node device rn name recording-health
~~~~~~~~~~~~~~~~ Recording Application  ~~~~~~~~~~~~~~~~
Recording Collection Time      : 2024-04-23 15:12:30 UTC
Recording CRC Errors           : 0
Recording Frame Length Errors  : 0
Recording Back Pressure Errors : 0
Recording Undersized Frames    : 0
Recording Frames 64B           : 0
Recording Frames 65-127B       : 0
Recording Frames 128-255B      : 0
Recording Frames 256-511B      : 0
Recording Frames 512-1023B     : 0
Recording Frames 1024-1522B    : 0
Recording Frames 1523-9522B    : 0
Recording Oversized Frames     : 0
Recording state                : Active

~~~~~~~~~~~~~~~~~~~~~~ Recording Instance  ~~~~~~~~~~~~~~~~~~~~~~
Core Disk    Dropped Packets Total Packets Collection Start Time
----|-------|---------------|-------------|----------------------|
1    primary 0               204           2024-04-23 15:00:34 UTC
```

```
C1> show recorder-node device rn name recording-health details
~~~~~~~~~~~~~~~~ Recording Application  ~~~~~~~~~~~~~~~~
Recording Collection Time      : 2024-04-23 15:12:30 UTC
Recording CRC Errors           : 0
Recording Frame Length Errors  : 0
Recording Back Pressure Errors : 0
Recording Undersized Frames    : 0
Recording Frames 64B           : 0
Recording Frames 65-127B       : 0
Recording Frames 128-255B      : 0
Recording Frames 256-511B      : 0
Recording Frames 512-1023B     : 0
Recording Frames 1024-1522B    : 0
Recording Frames 1523-9522B    : 0
Recording Oversized Frames     : 0
Recording state                : Active
Recording state Details        : Recording traffic.

~~~~~~~~~~~~~~~~~~~~~~ Recording Instance  ~~~~~~~~~~~~~~~~~~~~~~
Core Disk    Dropped Packets Total Packets Collection Start Time
----|-------|---------------|-------------|----------------------|
1    primary 0               204           2024-04-23 15:00:34 UTC
```

```
C1> show recorder-node device rn name recording-health
~~~~~~~~~~~~~~~~ Recording Application  ~~~~~~~~~~~~~~~~
Recording Collection Time      : 2024-04-23 15:12:30 UTC
Recording CRC Errors           : 0
Recording Frame Length Errors  : 0
Recording Back Pressure Errors : 0
Recording Undersized Frames    : 0
Recording Frames 64B           : 0
Recording Frames 65-127B       : 0
Recording Frames 128-255B      : 0
Recording Frames 256-511B      : 0
Recording Frames 512-1023B     : 0
Recording Frames 1024-1522B    : 0
Recording Frames 1523-9522B    : 0
Recording Oversized Frames     : 0
Recording state                : Active

~~~~~~~~~~~~~~~~~~~~~~ Recording Instance  ~~~~~~~~~~~~~~~~~~~~~~
Core Disk    Dropped Packets Total Packets Collection Start Time
----|-------|---------------|-------------|----------------------|
1    backup  0               204           2024-04-23 15:00:34 UTC
```

```
C1> show recorder-node device rn name recording-health details
~~~~~~~~~~~~~~~~ Recording Application  ~~~~~~~~~~~~~~~~
Recording Collection Time      : 2024-04-23 15:12:30 UTC
Recording CRC Errors           : 0
Recording Frame Length Errors  : 0
Recording Back Pressure Errors : 0
Recording Undersized Frames    : 0
Recording Frames 64B           : 0
Recording Frames 65-127B       : 0
Recording Frames 128-255B      : 0
Recording Frames 256-511B      : 0
Recording Frames 512-1023B     : 0
Recording Frames 1024-1522B    : 0
Recording Frames 1523-9522B    : 0
Recording Oversized Frames     : 0
Recording state                : Active
Recording state Details        : Recording traffic.

~~~~~~~~~~~~~~~~~~~~~~ Recording Instance  ~~~~~~~~~~~~~~~~~~~~~~
Core Disk    Dropped Packets Total Packets Collection Start Time
----|-------|---------------|-------------|----------------------|
1    backup  0               204           2024-04-23 15:00:34 UTC
```

The **show recorder-node warnings** command includes nonactive RN warnings if any exist.

```
C1> show recorder-node warnings
~ RAID Health Warning(s) ~
None.
~ Low Disk Space Warning(s) ~
None.
~ Full Disk Warning(s) ~
None.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Non Active Recorder Node Warning(s) ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~
# Recorder Node Message
-|-------------|-------------------------------------------------------------------------------------
--|
1 rn1           Inactive Recording State: Ready - Ready to record, no recordable traffic received.
```

Appendix C

DMF Controller in Microsoft Azure

The DANZ Monitoring Fabric (DMF) Controller in Azure feature supports the operation of the Arista Networks DMF Controller on the Microsoft Azure platform and uses the Azure CLI or the Azure portal to launch the Virtual Machine (VM) running the DMF Controller.

The DMF Controller in Azure feature enables the registration of VM deployments in Azure and supports **`auto-firstboot`** using Azure **`userData`** or **`customData`**.

C.1 Configuration

Configure Azure VMs **auto-firstboot** using **customData** or **userData**. There is no data merging from these sources, so provide the data via **customData** or **userData**, but not both.

Arista Networks recommends using **customData** as it provides a better security posture because it is available only during VM provisioning and requires **sudo** access to mount the virtual CDROM.

userData is less secure because it is available via Instance MetaData Service (IMDS) after provisioning and can be queried from the VM without any authorization restrictions.

If **sshKey** is configured for the admin account during Azure VM provisioning along with **auto-firstboot** parameters, then it is also configured for the **admin** user of DMF controllers.

The following table lists details of the firstboot parameters for the **auto-firstboot** configuration.

Firstboot Parameters - Required Parameters

Key	Description	Valid Values
admin_password	This is the password to set for the admin user. When joining an existing cluster node this will be the admin-password for the existing cluster node.	string
recovery_password	This is the password to set for the recovery user.	string

Additional Parameters

Key	Description	Required	Valid Values	Default Value
hostname	This is the hostname to set for the appliance.	no	string	Configured from Azure Instance Metadata Service
cluster_name	This is the name to set for the cluster.	no	string	Azure-DMF-Cluster
cluster_to_join	This is the IP which firstboot will use to join an existing cluster. Omitting this parameter implies that the firstboot will create a new cluster.  **Note:** If this parameter is present ntp-servers, cluster-name, and cluster-description will be ignored. The existing cluster node will provide these values after joining.	no	IP Address String	
cluster_description	This is the description to set for the cluster.	no	string	

Networking Parameters

Key	Description	Required	Valid Values	Default Value
ip_stack	What IP protocols to set up for the appliance management NIC.	no	enum: ipv4, ipv6, dual-stack	ipv4
ipv4_method	Setup IPv4 for the appliance management NIC.	no	enum: auto, manual	auto
ipv4_address	The static IPv4 address used for the appliance management NIC.	only if ipv4-method is set to manual	IPv4 Address String	
ipv4_prefix_length	The prefix length for the IPv4 address subnet to use for the appliance management NIC.	only if ipv4-method is set to manual	0..32	
ipv4_gateway	The static IPv4 gateway to use for the appliance management NIC.	no	IPv4 Address String	
ipv6_method	Set up IPv6 for the appliance management NIC.	no	enum: auto, manual	auto
ipv6_address	The static IPv6 address to use for the appliance management NIC.	only if ipv6-method is set to manual	IPv6 Address String	
ipv6_prefix_length	The prefix length for the IPv6 address subnet to use for the appliance management NIC.	only if ipv6-method is set to manual	0..128	
ipv6_gateway	The static IPv6 gateway to use for the appliance management NIC.	no	IPv6 Address String	
dns_servers	The DNS servers for the cluster to use	no	List of IP address strings	
dns_search_domains	The DNS search domains for the cluster to use.	no	List of the host names or FQDN strings	
ntp_servers	The NTP servers for the cluster to use.	no	List of the host names of FQDN strings `0.bigswitch.pool.ntp.org` **`1.bigswitch.pool.ntp.org`** **`2.bigswitch.pool.ntp.org`** **`3.bigswitch.pool.ntp.org`**	

Examples

```
{
    "admin_password": "admin_user_password",
    "recovery_password": "recovery_user_password"
}
```

Full List of Parameters

```
{
    "admin-password": "admin_user_password",
    "recovery_password": "recovery_user_password",
    "hostname": "hostname",
    "cluster_name": "cluster name",
    "cluster_description": "cluster description",
    "ip_stack": "dual-stack",
    "ipv4_method": "manual",
    "ipv4_address": "10.0.0.3",
    "ipv4_prefix-length": "24",
    "ipv4_gateway": "10.0.0.1",
    "ipv6_method": "manual",
    "ipv6_address": "be:ee::1",
    "ipv6_prefix-length": "64",
    "ipv6_gateway": "be:ee::100",
    "dns_servers": [
    "10.0.0.101",
    "10.0.0.102"
    ],
    "dns_search_domains": [
    "dns-search1.com",
    "dns-search2.com"
    ],
    "ntp_servers": [
    "1.ntp.server.com",
    "2.ntp.server.com"
    ]
}
```

C.2 Syslog Messages

- There are three possible failure modes:
  - VM fails Azure registration.
  - **`auto-firstboot`** fails due to a transient error or bug.
  - **`auto-firstboot`** parameter validation fails.
- These failures can be debugged by accessing the **`firstboot`** logs after manually booting the VM or logging via the **`recovery`** user on Azure serial console:
  - Azure DMF Controller VMs can be accessed via **`ssh`** login after successful **`firstboot`**:

    ```
    dmf-controller-0-vm> enable; configure;
    dmf-controller-0-vm> show logging syslog | grep 'floodlight-autofirstboot'
    ```

  - For debugging parameter validation errors, access the parameter validation results:

    ```
    dmf-controller-0-vm> show firstboot parameter-validation
    ```

- Accessing logs via the **`recovery`** user on Azure serial console. The following output is an example log for missing a required **`firstboot`** parameter – **`admin_password`**:

```
Log in as 'admin' to configure

controller login: recovery
recovery@controller:~$ cat /var/log/floodlight/firstboot/firstboot.log
...
2024-06-17 17:09:09,982 autofirstboot: CRITICAL [main] Uncaught exception
Traceback (most recent call last):
File "/usr/bin/floodlight-autofirstboot", line 11, in <module>
load_entry_point('firstboot==0.1.0', 'console_scripts', 'floodlight-autofirstboot')()
File "/usr/share/floodlight/firstboot/firstboot/autofirstboot.py", line 93, in main
params, plugin = get_params(plugins)
File "/usr/share/floodlight/firstboot/firstboot/autofirstboot.py", line 44, in get_params
params = plugin.get_firstboot_params()
File "/usr/share/floodlight/firstboot/firstboot/cloud_plugins/azure.py", line 75, in
 get_firstboot_params
return FirstbootParams(**firstboot_param_dict)
TypeError: __init__() missing 1 required positional argument: 'admin_password'
```

C.3 Troubleshooting

- If a DMF Controller VM cannot be accessed via ssh login, the auto-firstboot has probably failed.
- The DMF Controller VMs must be recreated on Azure for any transient failure for VM registration with Azure or for auto-firstboot to occur.
- The DMF Controller VMs can also be configured manually for firstboot via the Azure serial console.

C.4 Limitations

The following limitations apply to the DANZ Monitoring Fabric (DMF) Controller in Microsoft Azure.

- There is no support for any features specific to Azure-optimized Ubuntu Linux, including Accelerated Networking.
- The DMF Controllers in Azure are only supported on Gen-1 VMs.
- The DMF Controllers in Azure do not support adding the virtual IP address for the cluster.
- There is no support for capture interfaces in Azure.
- DMF ignores the Azure username and password fields.
- There is no support for static IP address assignment that differs from what is configured on the Azure NIC.
- The DMF Controllers are rebooted if the static IP on the NIC is updated.
- Switches are supported in L3 ZTN mode only.

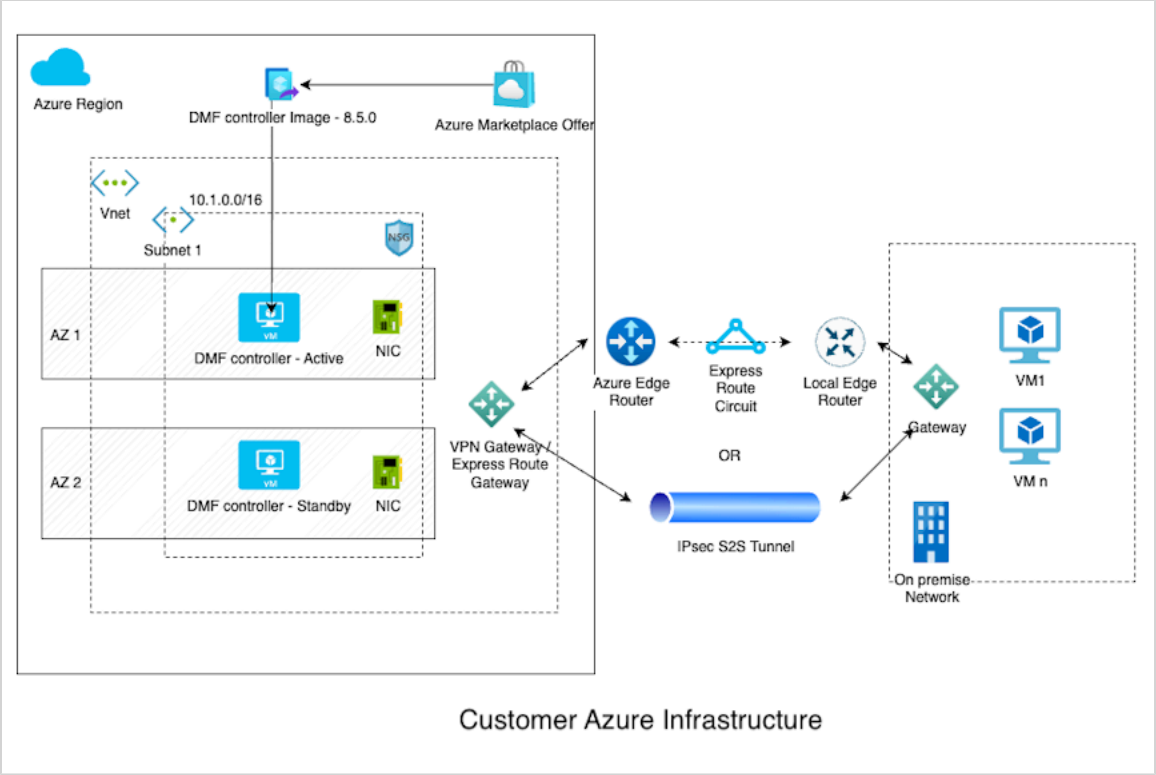
C.5 Resources

Please refer to the following resources for more information.

- Azure User Data: https://learn.microsoft.com/en-us/azure/virtual-machines/user-data
- Azure Custom Data: https://learn.microsoft.com/en-us/azure/virtual-machines/custom-data
- Azure Serial Console: https://learn.microsoft.com/en-us/troubleshoot/azure/virtual-machines/windows/serial-console-overview
- Azure Gen1 vs Gen2 VMs: https://learn.microsoft.com/en-us/azure/virtual-machines/generation-2
- Azure Optimized Ubuntu Linux Features: https://ubuntu.com/blog/microsoft-and-canonical-increase-velocity-with-azure-tailored-kernel
- Azure NIC Assignment Behavior: https://learn.microsoft.com/en-us/troubleshoot/azure/virtual-machines/reset-network-interface-azure-linux-vm
- DMF Hardware and Software Requirements: https://www.arista.com/en/hcl-dmf/hcl-supported-software-controller-hardware-and-switch-hardware

C.6 Diagrams

Figure C-1: Customer Azure Infrastructure



Appendix D

DMF Controllers in Google Cloud VMware Engine

The DANZ Monitoring Fabric (DMF) Controller in Google Cloud VMware Engine (GCVE) supports the operation of the Arista Networks DMF Controller on the GCVE platform and uses the vCenter portal to launch the Virtual Machine (VM) running the DMF Controller.

The DMF Controller in GCVE enables the registration of VM deployments in the vCenter environment and supports **`auto-firstboot`** when deployed using the **Deploy OVF Template** wizard in the vCenter.

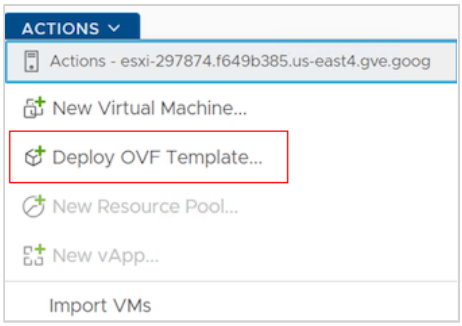
Configuration

GCVE portal provides the vCenter server IP address with the credential details. Log in to the vCenter using these credentials to deploy the Controller VM using the Deploy OVF Template wizard in the vCenter.

The following describes using the wizard to deploy the Controller VM.

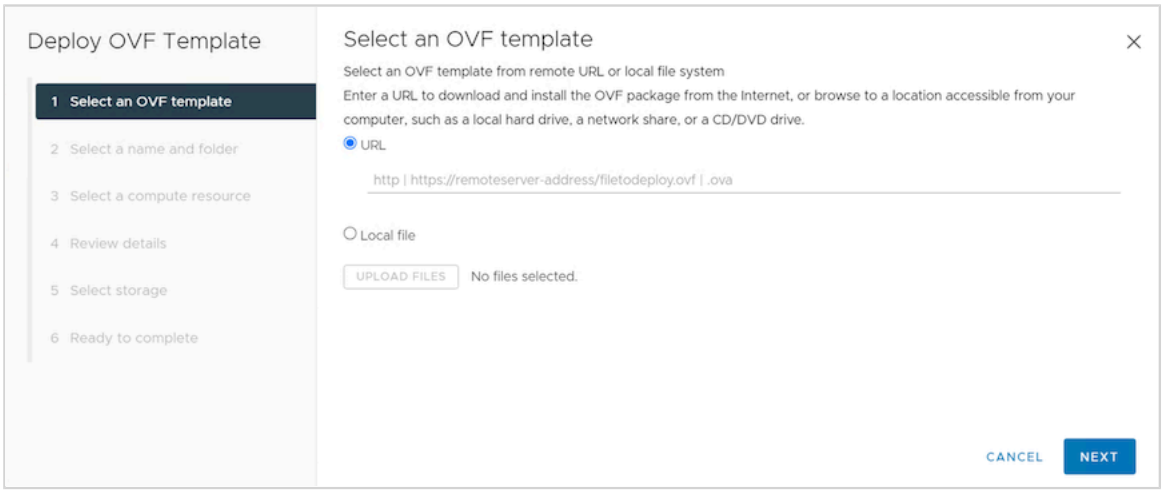
1. In VMware vCenter, navigate to the **Actions** menu and select **Deploy OVF Template**.

   Figure D-1: Deploy OVF Template



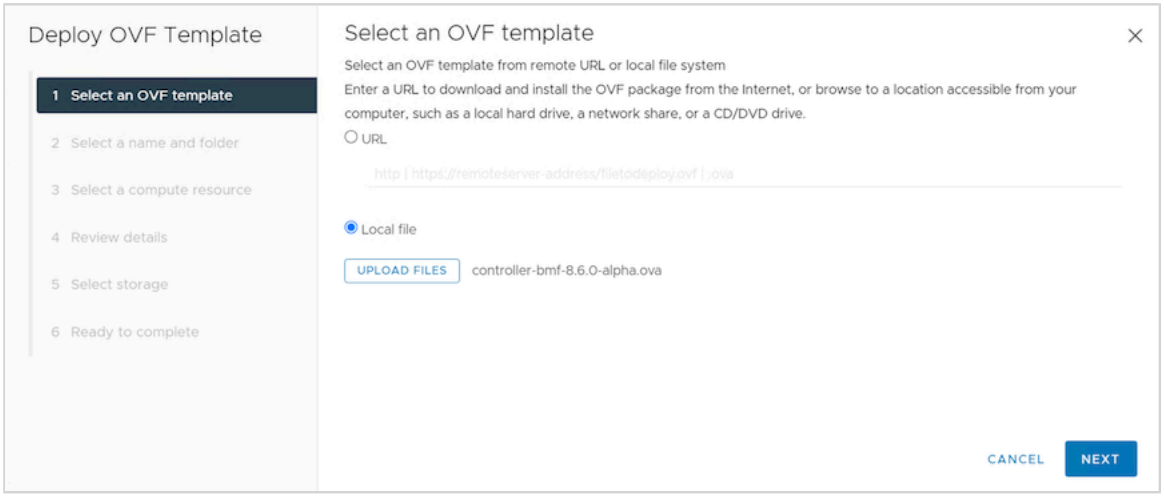
2. The **Deploy OVF Template** wizard appears.

Figure D-2: Deploy OVF Template Wizard



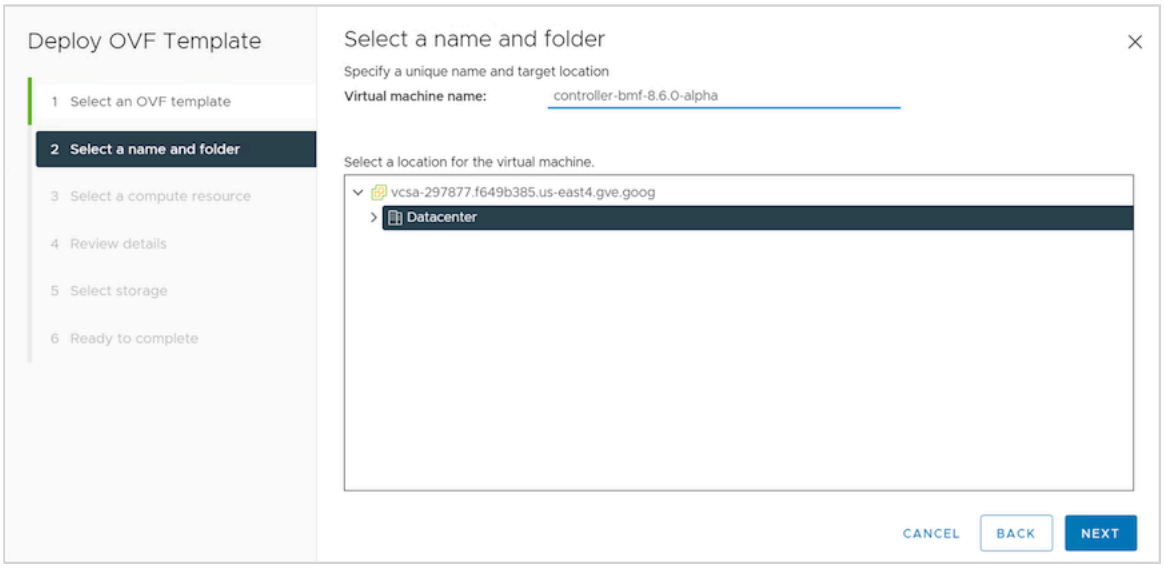
3. Enter the **URL** for the OVA file. If the OVA file is available locally on the machine, choose the **Local file** option and select **Upload Files**.

Figure D-3: Enter URL or Local File



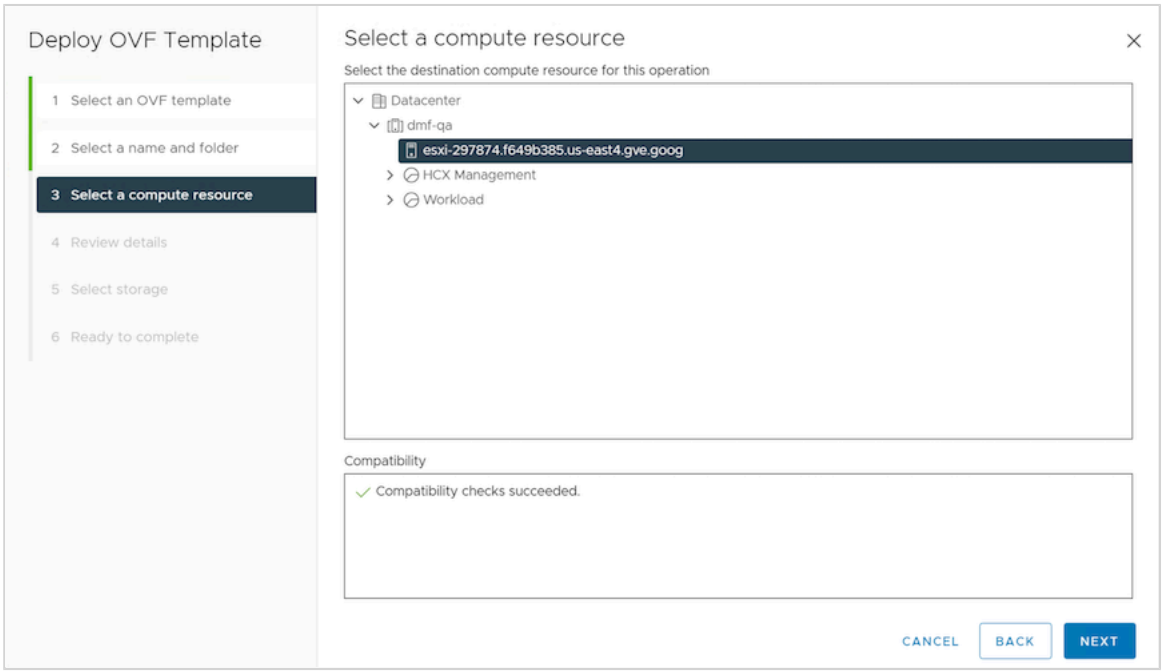
4. Provide the **Virtual machine name** and select its **location**.

Figure D-4: Virtual Machine Name



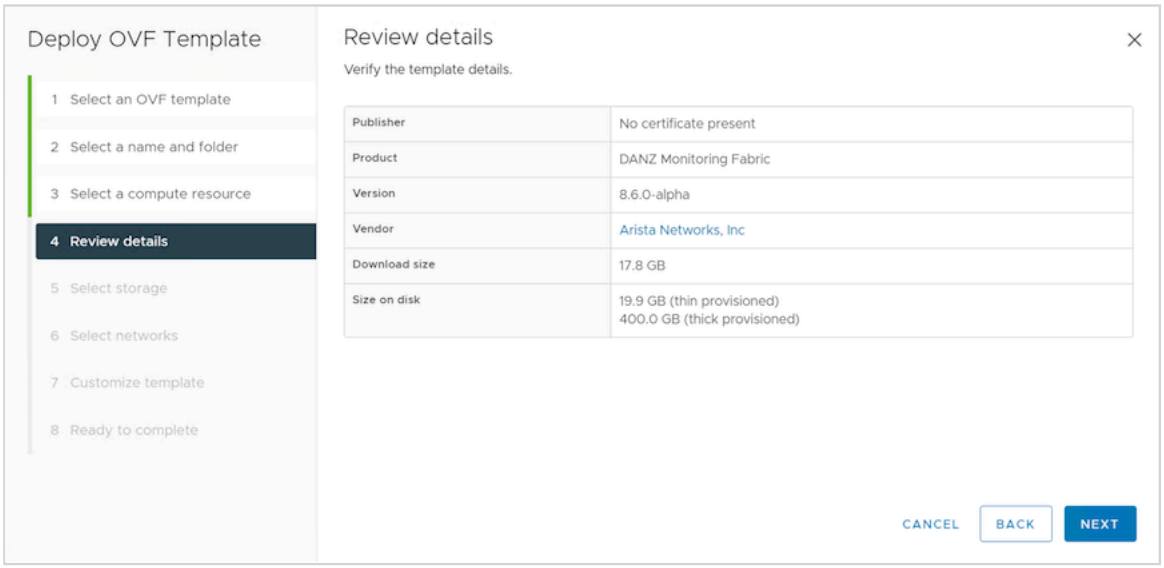
5. Select the **destination compute resource**.

Figure D-5: Compute Resource



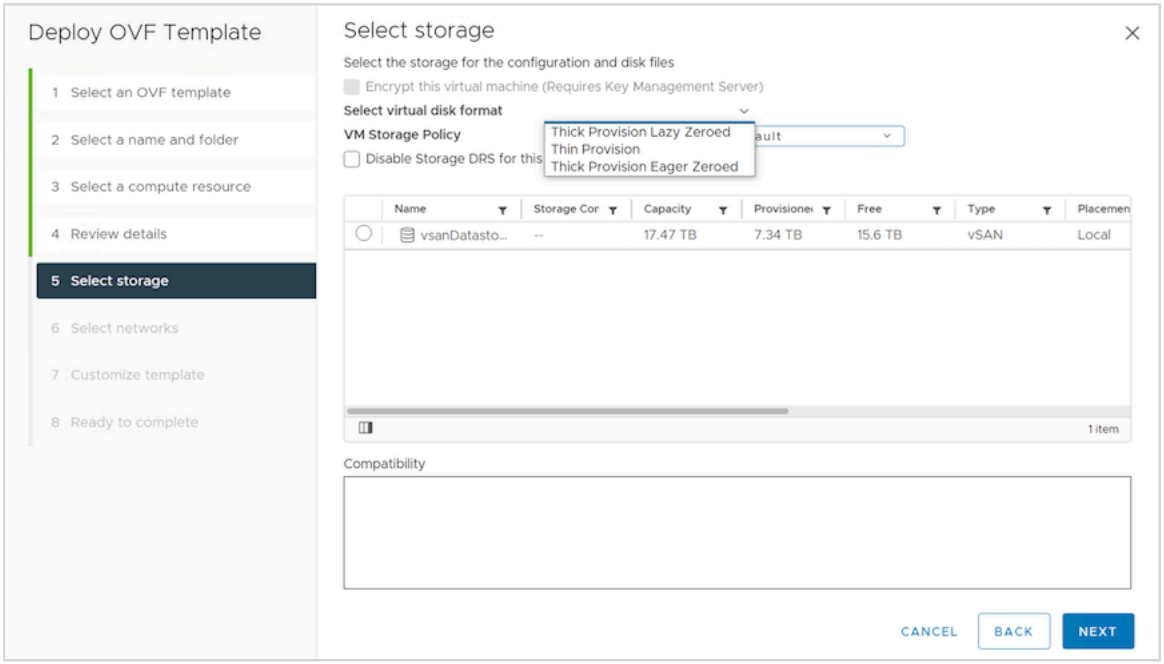
6. **Review details** summarize the OVA file details.

Figure D-6: Review Details



7. Provide the **storage details** for the VM.

Figure D-7: Storage Details



8. As illustrated in the following **Customize template** sections, **autofirstboot** parameters appear. Use these parameters to configure the VM to make it ready after the OVF deployment. There are 19 settings; however, some are mandatory, while others are optional.

Figure D-8: Customize Template Auto First Boot – 19 Settings

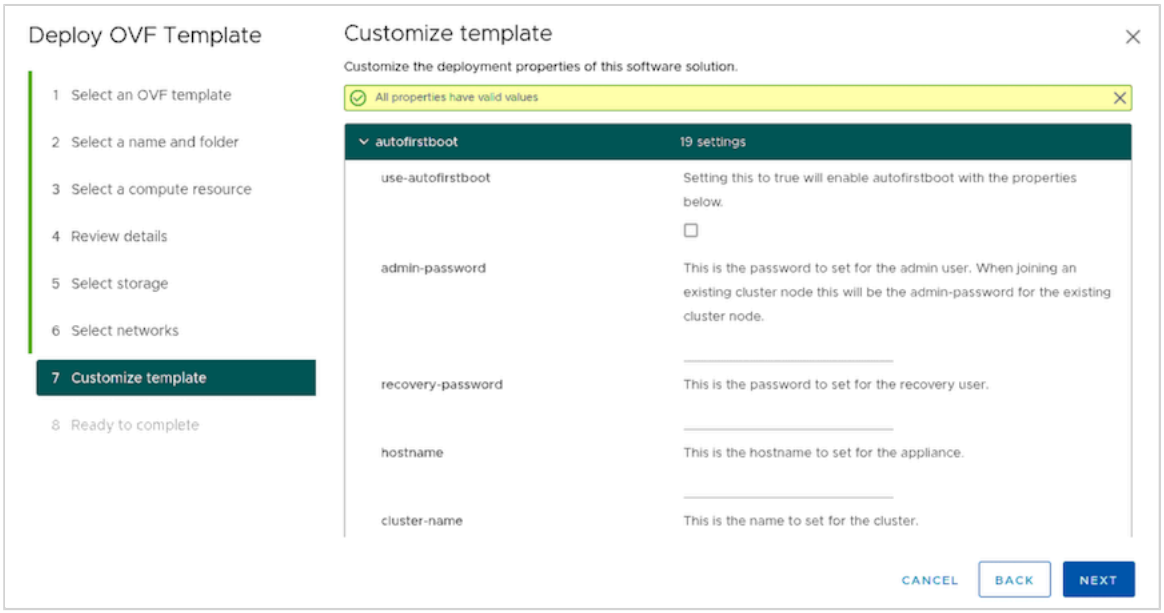


Figure D-9: Customize Template Auto First Boot

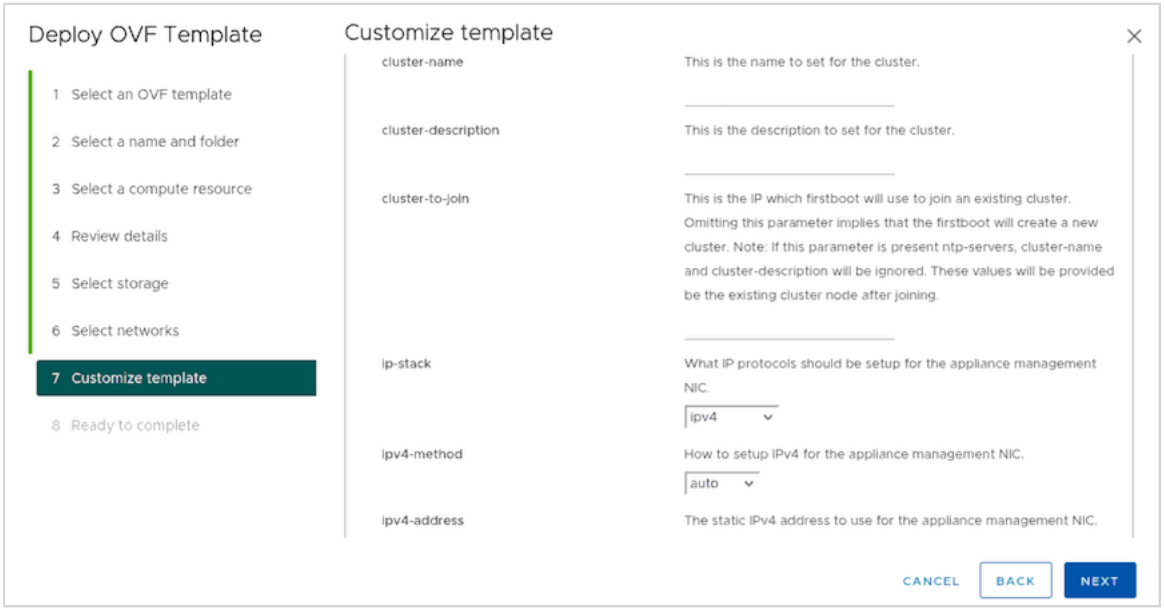


Figure D-10: Customize Template Auto First Boot

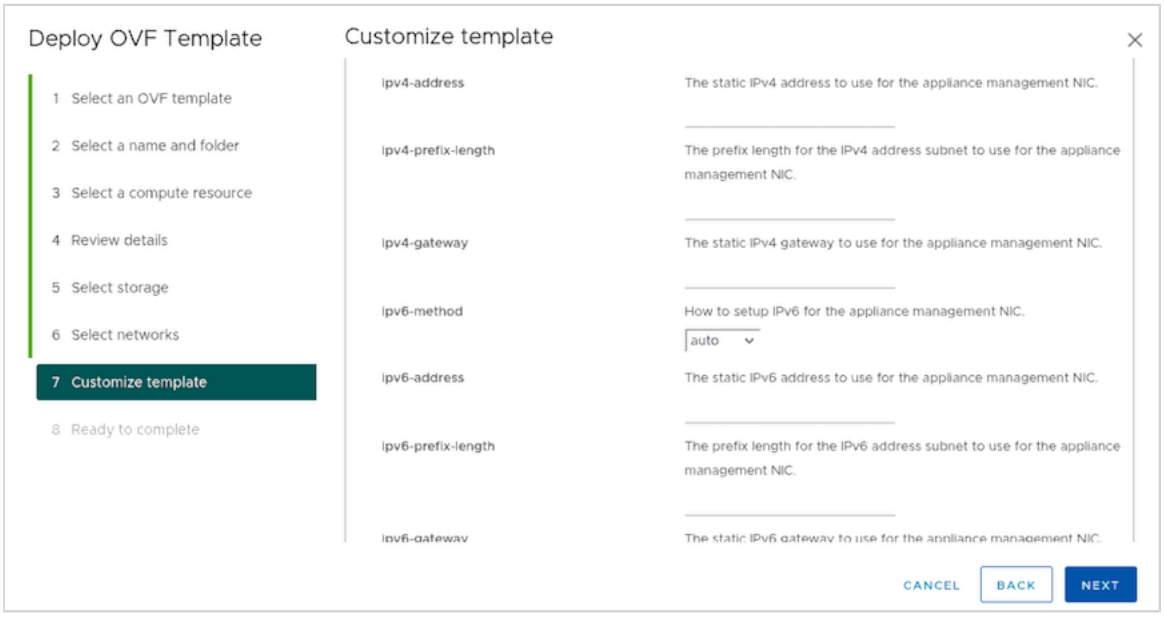
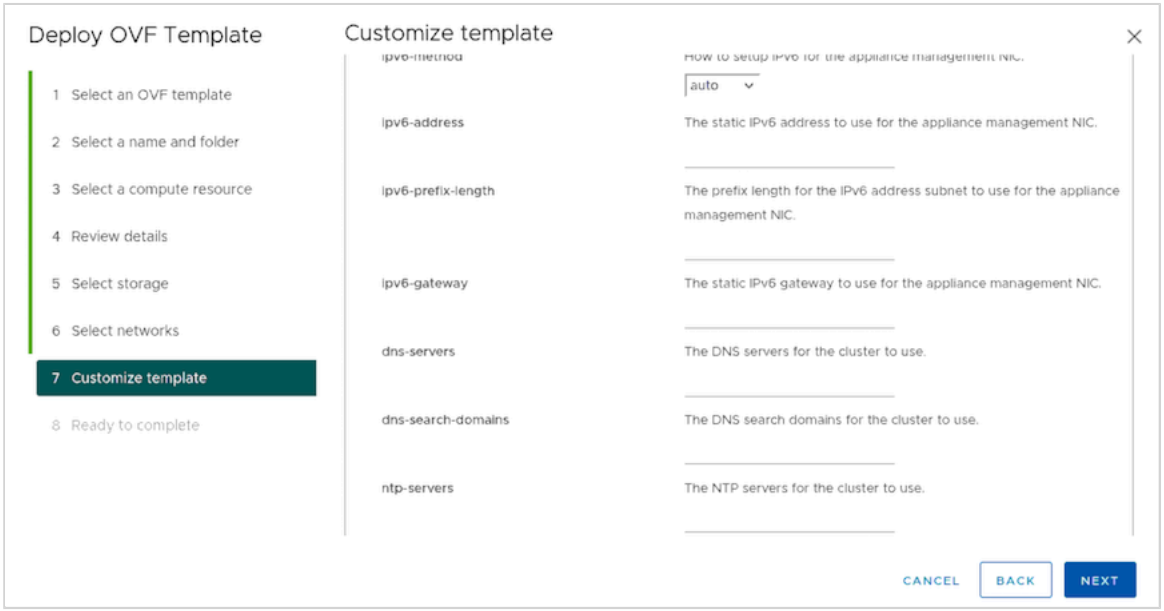
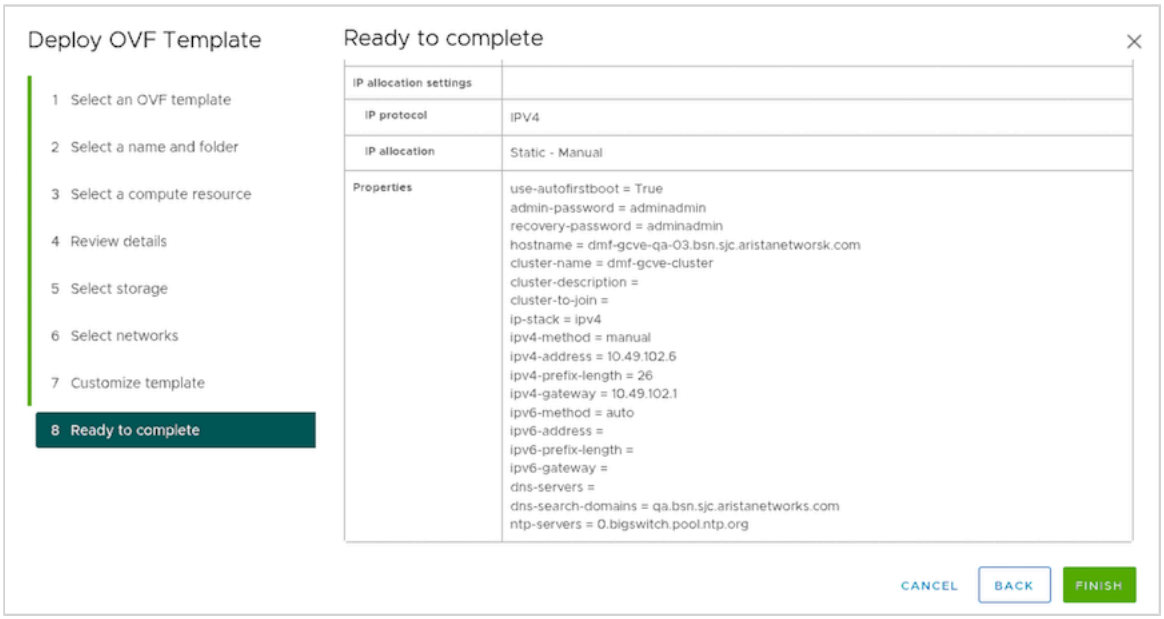


Figure D-11: Customize Template Auto First Boot



9. Review the **Ready to complete** firstboot parameters and select **Finish** to enter the configuration. If changes are required, select **Back** and make those changes.

Figure D-12: Ready To Complete



D.1 Firstboot Parameters

The following table lists the firstboot parameters for the **`auto-firstboot`** configuration.

Required Parameters

Key	Description	Valid Values
`admin_password`	This is the password to set for the admin user. When joining an existing cluster node this will be the admin-password for the existing cluster node.	string
`recovery_password`	This is the password to set for the recovery user.	string

Additional Parameters

Key	Description	Required	Valid Values	Default Value
`hostname`	This is the hostname to set for the appliance.	no	string	
`cluster_name`	This is the name to set for the cluster.	no	string	
`cluster_to_join`	This is the IP which firstboot will use to join an existing cluster. Omitting this parameter implies that the firstboot will create a new cluster. Note: If this parameter is present ntp-servers, cluster-name, and cluster-description will be ignored. The existing cluster node will provide these values after joining.	no	IP Address String	
`cluster_description`	This is the description to set for the cluster.	no	string	

Networking Parameters

Key	Description	Required?	Valid Values	Default Value
ip_stack	What IP protocols should be set up for the appliance management NIC.	no	enum: ipv4, ipv6, dual-stack	ipv4
ipv4_method	How to setup IPv4 for the appliance management NIC.	no	enum: auto, manual	auto
ipv4_address	The static IPv4 address used for the appliance management NIC.	only if ipv4-method is set to manual	IPv4 Address String	
ipv4_prefix_length	The prefix length for the IPv4 address subnet to use for the appliance management NIC.	only if ipv4-method is set to manual	0..32	
ipv4_gateway	The static IPv4 gateway to use for the appliance management NIC.	no	IPv4 Address String	
ipv6_method	How to set up IPv6 for the appliance management NIC.	no	enum: auto, manual	auto
ipv6_address	The static IPv6 address to use for the appliance management NIC.	only if ipv6-method is set to manual	IPv6 Address String	
ipv6_prefix_length	The prefix length for the IPv6 address subnet to use for the appliance management NIC.	only if ipv6-method is set to manual	0..128	
ipv6_gateway	The static IPv6 gateway to use for the appliance management NIC.	no	IPv6 Address String	
dns_servers	The DNS servers for the cluster to use	no	List of IP address strings	
dns_search_domains	The DNS search domains for the cluster to use.	no	List of the hostnames or FQDN strings	
ntp_servers	The NTP servers for the cluster to use.	no	List of the hostnames of FQDN strings: 0.bigswitch.pool.ntp.org 1.bigswitch.pool.ntp.org 2.bigswitch.pool.ntp.org 3.bigswitch.pool.ntp.org	

D.2 Post Installation Verification

1. Access a DMF Controller VM using an SSH login session.
2. Verify the Controller clusters have formed using the **`show controller`** command.

```
GCVE-DMF-860-1# show controller
Cluster Name           : GCVE-DMF-860-1
Cluster Description    : GCVE DMF CONTROLLER
Cluster Virtual IP     : 10.100.189.231
Redundancy Status      : redundant
Last Role Change Time  : 2024-05-21 14:23:38.917000 PDT
Failover Reason        : Changed connection state: connected to node 21773
Device deployment mode : pre-configured
Cluster Uptime         : 1 day, 3 hours
# Hostname       @ State   Uptime
-|--------------|-|-------|-------------------|
1 GCVE-DMF-860-1 * active  5 hours, 6 minutes
2 GCVE-DMF-860-2   standby 15 minutes, 11 secs
GCVE-DMF-860-1#
```

3. Use the **`show version`** command to verify the installed version is correct.

```
GCVE-DMF-860-1# show version
Controller Version : DANZ Monitoring Fabric 8.6.0 (bmf/dmf-8.6.0 #10)
GCVE-DMF-860-1#
```

D.3 Troubleshooting

The following are possible failure modes:

- **`auto-firstboot`** fails due to a transient error or bug.
- **`auto-firstboot`** parameter validation fails.

To debug these failures, connect to the VM using the console, log in as user `recovery`, and use the following command to review the **`firstboot`** logs.

```
recovery@dmf-controller-0-vm:~$ less /var/log/floodlight/firstboot/firstboot.log
```

To debug parameter validation errors, access the parameter validation results using the following command.

```
recovery@dmf-controller-0-vm:~$ less /var/lib/floodlight/firstboot/validation-results.json
```

D.4 Limitations

- There is no support for capture interfaces in GCVE.

Appendix E

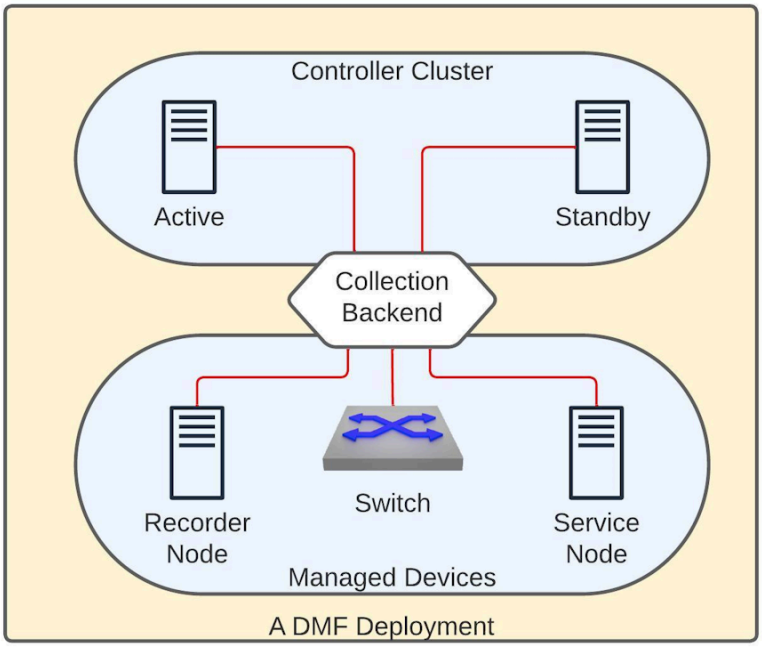
Telemetry Collector

A DANZ Monitoring Fabric (DMF) consists of a pair of Controllers, switches, and managed appliances. The Telemetry Collector feature centrally retrieves the infrastructure metrics (interface counters, CPU usage, etc.) affiliated with all these devices from the controllers using its REST API.

Storage Health for CVA 7.1.0

Deployment

Figure E-1: DMF Deployment



The diagram above shows a DMF deployment with an Active/Standby Controller cluster. In this environment, each Controller collects metrics from all the devices it manages and the controller nodes. Each Controller establishes a gNMI connection to all the devices and the other Controller in a fabric to collect telemetry streams. gNMI is a gRPC-based protocol to configure and access states on network devices. The REST API exposes this information in the `/api/v1/data/controller/telemetry/data` subtree. The Controller exposes several metrics about CPU, memory, disk utilization, interface counters, and sensor states of all devices in a fabric. All metrics are fetched at a 10-second frequency except those associated with the sensors, which are collected every minute.

Configuration

No additional configuration is necessary on the DMF Controller to enable the metric collection. However, to read these metrics, the user must be an admin user or configured with the privilege, category:TELEMETRY. To set the telemetry permission for a custom group and associate the user with this group, use the following commands:

```
dmf-controller(config)# group group_name
dmf-controller(config)# permission category:TELEMETRY privilege read-only
dmf-controller(config)# associate user username
```

Connection Status

The Controller uses the gNMI protocol to collect telemetry data from all devices. DMF reports the status of these connections in both the REST API and the CLI.

REST API

The REST API subtree `/api/v1/data/controller/telemetry/connection` reports the telemetry connection state. The API Schema browser of the GUI provides more details.

CLI

The following **show** commands display the connection details. All these commands support filtering the output with a device name.

```
show telemetry connection device-name | all
show telemetry connection device-name | all details
show telemetry connection device-name | all last-failure
```

The **show telemetry connection device-name all** command shows the state and the latest state change of the connection between the Controller and the devices.

```
dmf-controller# show telemetry connection all
# Name                          State Last state change
-|-----------------------------|-----|-----------------------------|
1 c2                            ready 2023-11-10 13:10:02.718000 UTC
2 core2                         ready 2023-11-10 13:22:56.311000 UTC
<snip>
```

The **show telemetry connection device-name all** command displays the state and the latest state change of the connection between the Controller and the devices.

```
dmf-controller# show telemetry connection all details
# Name  State Last state change              Target              Connection type Last message time
-|-----|-----|-----------------------------|-------------------|---------------|-----------------------------|
1 core2 ready 2023-11-10 13:22:56.311000 UTC 10.243.255.102:6030 clear-text      2023-11-10 13:59:35.437000 UTC
<snip>
```

The **show telemetry connection device-name all last-failure** command displays more details about a connection failure. The time of the latest failure and the potential reason appear in the output. If the connection is still in the failed state, this output also shows when the next reconnection is attempted.

```
dmf-controller# show telemetry connection all last-failure
# Name  Fail time                        Fail type       Root cause                 Next retry in
-|-----|-----------------------------|---------------|--------------------------|-------------|
1 core2 2023-11-10 13:19:34.237000 UTC unavailable     UNAVAILABLE: io exception 0
<snip>
```

Limitations

- Software interfaces (for example, loopback, bond, and management) do not report counters for broadcast and unicast packets.
- The reported interface names are the raw physical interface name (e.g., **et1**) rather than the user-configured name associated with the role of an interface (e.g., **filter1**).
- Resetting the interface counter does not affect the counter values stored at the `/telemetry/data` path. The value monotonically increases and corresponds to the total count since the device was last powered up. This value only gets reset when rebooting the device.

Usage Notes

- DMF uses the configured name of a managed device (e.g., switch, recorder node, etc.) on the Controller as the value of the key name for the node device for all the metrics corresponding to it. In the case of a Controller, DMF uses the configured hostname as the key. Thus, these names must be unique in a specific DMF deployment.
- The possibility exists that metrics are not collected from a device for a short period. This data gap may happen when rebooting the device or when the Controllers experience a failover event.
- If the gNMI connection between the Controller and the device is interrupted, the Controller attempts a new connection after 2 minutes. The retry timeout for a subsequent connection attempt increases exponentially and can go up to 120 minutes. Upon a successful reconnection, this timeout value resets to 2 minutes.
- There might be gNMI warning messages in the floodlight log during certain events, e.g., when first adding a device or it is reloading. Ignore these messages.
- This feature enables an OpenConfig agent on switches running EOS to collect telemetry.

Telemetry Availability

As a DMF fabric consists of different types of devices, the metrics of each vary. The following outlines the metrics collected from each device type by the Controller and typically made available over its REST API. However, some specific platforms or hardware might not report a particular metric. For brevity, the following list mentions the leaves that can correspond to a metric. For more details, use the API Schema browser of the GUI.

```
telemetry
+-- data
     +-- device
         +-- interface
         |  +-- oper-status                Ctrl, SWL, EOS, SN, RN
         |  +-- counters
         |     +-- in-octets               Ctrl, SWL, EOS, SN, RN
         |     +-- in-pkts                 Ctrl, SWL, EOS, SN, RN
         |     +-- in-unicast-pkts         Ctrl, SWL, EOS, SN, RN
         |     +-- in-broadcast-pkts       Ctrl, SWL, EOS, SN, RN
         |     +-- in-multicast-pkts       Ctrl, SWL, EOS, SN, RN
         |     +-- in-discards             Ctrl, SWL, EOS, SN, RN
         |     +-- in-errors               Ctrl, SWL, EOS, SN, RN
         |     +-- in-fcs-errors           Ctrl, SWL, EOS, SN, RN
         |     +-- out-octets              Ctrl, SWL, EOS, SN, RN
         |     +-- out-pkts                Ctrl, SWL, EOS, SN, RN
         |     +-- out-unicast-pkts        Ctrl, SWL, EOS, SN, RN
         |     +-- out-broadcast-pkts      Ctrl, SWL, EOS, SN, RN
         |     +-- out-multicast-pkts      Ctrl, SWL, EOS, SN, RN
         |     +-- out-discards            Ctrl, SWL, EOS, SN, RN
         |     +-- out-errors              Ctrl, SWL, EOS, SN, RN
         +-- cpu
         |  +-- utilization                Ctrl, SWL, EOS, SN, RN
         +-- memory
         |  +-- total                      Ctrl, SWL, SN, RN
         |  +-- available                  Ctrl, SWL, EOS, SN, RN
         |  +-- utilized                   Ctrl, SWL, EOS, SN, RN
         +-- sensor
         |  +-- fan
         |  |  +-- oper-status             Ctrl, SWL, EOS, SN, RN
         |  |  +-- rpm                     Ctrl, SWL, EOS, SN, RN
         |  |  +-- speed                   SWL
         |  +-- power-supply
         |  |  +-- oper-status             Ctrl, SWL, EOS, SN, RN
         |  |  +-- capacity                EOS
         |  |  +-- input-current           Ctrl, SWL, EOS, SN, RN
         |  |  +-- output-current          SWL, EOS
         |  |  +-- input-voltage           Ctrl, SWL, EOS, SN, RN
         |  |  +-- output-voltage          SWL, EOS
         |  |  +-- input-power             Ctrl, SWL, SN, RN
         |  |  +-- output-power            SWL, EOS
         |  +-- thermal
         |     +-- oper-status             Ctrl, SWL, SN, RN
         |     +-- temperature             Ctrl, SWL, EOS, SN, RN
         +-- mount-point
         |  +-- size                       Ctrl, SWL, SN, RN
         |  +-- available                  Ctrl, SWL, SN, RN
         |  +-- utilized                   Ctrl, SWL, SN, RN
         |  +-- usage-percentage           Ctrl, SWL, SN, RN
         +-- control-group
         +-- memory                     Ctrl, SWL, SN, RN
         +-- cpu                        Ctrl, SWL, SN, RN

* Ctrl = Controller, SWL = A switch running SwichLight OS,
EOS = A switch running Arista EOS, SN = Service Node, RN = Recorder Node
```

E.1 Storage Health for CVA 7.1.0

Storage Health for CVA 7.1.0 exposes metrics and health status of storage devices on controllers and all managed nodes, but not switches. Metrics and health status are updated every minute and exposed through the Telemetry Collector.

The feature is compatible with all the supported platforms. However, the exact set of metrics may vary depending on the hardware configuration.

The appliance does not require additional configuration. The metrics are exposed using the Telemetry Collector, and reading the collected data requires the same permissions.

REST API

REST `subtree /api/v1/data/controller/os/status/storage` reports the storage configuration of the current appliance - it contains attributes of available physical drives, RAID controllers, and virtual drives. REST subtree `/api/v1/data/controller/telemetry/data/device/storage` reports the storage devices' metrics and health status. The API Schema browser of the GUI (/schema-browser URL on the Controller) provides more details.

Show Commands

The **`show hardware storage-configuration`** command, corresponding to the REST subtree `/api/v1/data/controller/os/status/storage`, lists storage devices and their attributes.

```
ctrl01> show hardware storage-configuration
Collection result : success
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Virtual Drives  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Virtual Drive ID                       Name OS Name  RAID  Controller ID             Internal ID
-|--------------------------------------|----|--------|-----|-------------------------|-----------|
1 scsi-362cea7f0c91726002f62b6e605537f47 os   /dev/sda RAID5 PERC_H330_Adapter/18I031B /c0/v0
2 scsi-362cea7f0c91726002f62e0cb115e191e data /dev/sdb RAID5 PERC_H330_Adapter/18I031B /c0/v1

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Physical Drives  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Physical Drive ID    Model        Serial Number     Controller ID         Internal ID
-|--------------------|------------|-------------|-------------------------|-----------|
1 HFS1T9-BA10A/0307O1F HFS1T9-BA10A 0307O1F       PERC_H330_Adapter/18I031B /c0/e32/s2
2 HFS1T9-BA10A/0307O1G HFS1T9-BA10A 0307O1G       PERC_H330_Adapter/18I031B /c0/e32/s0
3 HFS1T9-BA10A/0307O1S HFS1T9-BA10A 0307O1S       PERC_H330_Adapter/18I031B /c0/e32/s1
4 HFS1T9-BA10A/0307O8S HFS1T9-BA10A 0307O8S       PERC_H330_Adapter/18I031B /c0/e32/s3

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Controllers  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Controller ID             Model             Serial Number Internal ID
-|-------------------------|-----------------|-------------|-----------|
1 PERC_H330_Adapter/18I031B PERC H330 Adapter 18I031B       /c0
```

```
ctrl01> show hardware storage-configuration virtual-drive scsi-362cea7f0c91726002f62b6e605537f47
Virtual Drive ID : scsi-362cea7f0c91726002f62b6e605537f47
Controller ID    : PERC_H330_Adapter/18I031B
Internal ID      : /c0/v0
Name             : os
OS Name          : /dev/sda
RAID             : RAID5
~~~ Physical Drives ~~~
# Physical Drive ID
-|--------------------|
1 HFS1T9-BA10A/0307O1F
2 HFS1T9-BA10A/0307O1G
3 HFS1T9-BA10A/0307O1S
4 HFS1T9-BA10A/0307O8S
```

The Controller ID, Internal ID, RAID, and Name fields are empty for drives not managed by a RAID controller. The Name field is also empty if no name is assigned to the virtual drive within the RAID controller's configuration. The show hardware storage-health command prints the overall health status of the storage devices collected from all the DMF cluster nodes.

```
ctrl01> show hardware storage-health
~~~~~~~~~~~~~~~~~~~~~~ Device Summary ~~~~~~~~~~~~~~~~~~~~~~
# Device Status Collection result Collection error messages
-|------|------|-----------------|-------------------------|
1 ctrl01 ok     success
2 ctrl02 ok     success

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Virtual Drives ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Device Virtual Drive ID                       Status  Messages
-|------|-------------------------------------|-------|-------------------------------------------------------------------------------|
1 ctrl01 scsi-362cea7f0c91726002f62b6e605537f47 ok
2 ctrl01 scsi-362cea7f0c91726002f62e0cb115e191e ok
3 ctrl02 scsi-36d0946603abfc90022181eae0f6f9708 ok
3 ctrl02 scsi-36d0946603abfc90022181eae0f6f9708 warning Physical drive "HUS722/D1N42T" is in a suboptimal state

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Physical Drives  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~
# Device Physical Drive ID    Status Messages
-|------|--------------------|------|-------------------------------------------------------------------------------------------------------------------
--|
1 ctrl01 HFS1T9-BA10A/0307O1F ok
2 ctrl01 HFS1T9-BA10A/0307O1G ok
3 ctrl01 HFS1T9-BA10A/0307O1S ok
4 ctrl01 HFS1T9-BA10A/0307O8S ok
5 ctrl02 HUS722/D1N42T        error  SMART self-assessment failed, SMART attribute "Raw_Read_Error_Rate" (1) is below the threshold
6 ctrl02 HUS722/D94WR9        ok

~~~~~~~~~~~~~~~~~~~ Controllers ~~~~~~~~~~~~~~~~~~~
# Device Controller ID             Status Messages
-|------|-------------------------|------|--------|
1 ctrl01 PERC_H330_Adapter/18I031B ok
1 ctrl02 PERC_H330_Adapter/7AA004D ok
```

A Collection result of **error** under Device Summary indicates that gathering metrics was partially successful. In such a case, the health status of storage devices may not be accurate. The **Status** column in the Virtual Drives, Physical Drives, and Controllers tables indicates the overall health of the device:

- **OK** - no issues identified.
- **Warning** - the device is operational, but not at the full performance level.
- **Error** - the device has already failed or is deemed to fail soon.

Regarding warning and error statuses, further details are in the Messages column. The detailed metrics for physical drives are accessible using the **`show hardware storage-health device-id physical-drive drive-id`** command:

```
cont01> show hardware storage-health cont01 physical-drive HFS1T9-BA10A/0307O1F
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Drives  ~~~~~~~~~~~~~~~~~~~
```

E.1.1 Troubleshooting

- If functionality is hampered, the REST API and CLI report error collection status and messages describe the issue. Review the device's `floodlight.log` and `events.log` for further details about the issue.



Note: The log files do not report certain storage issues, such as a failing hard drive.

E.1.2 Considerations

- The feature relies on metrics reported by storage devices (drives and controllers). Therefore, the exact set of metrics, accuracy, and reliability may vary depending on the hardware.
- The feature only gathers metrics exposed by the devices. It does not perform any tests, including physical drive self-tests, which may provide additional data points for assessing storage device health.
- The feature is stateless. It does not track changes in metrics or changes in hardware configuration. However, metrics' dynamics can be tracked with external tooling, e.g., by exporting them to Prometheus (refer to https://www.arista.com/en/support/toi/dmf-8-5-0/18913-prometheus-endpoint-support-for-infrastructure-metrics).

Appendix F

DMF GUI REST API Inspector

Overview

DANZ Monitoring Fabric (DMF) 8.6 introduced a newly designed API Inspector available on all DMF UI pages. Previously, the former API Inspector was accessible by selecting the Dragonfly icon, and it was only available on some pages.

Launch the API Inspector

1. Hover over and select the **User** icon to see the drop-down menu.
2. Select **API Inspector** to open the API Inspector table.
3. The REST API Inspector table appears between the page content and the navigation bar. It will remain open when switching between pages.

4. To close it, select **API Inspector** or the **×** icon.

Figure F-1: Menu > API Inspector

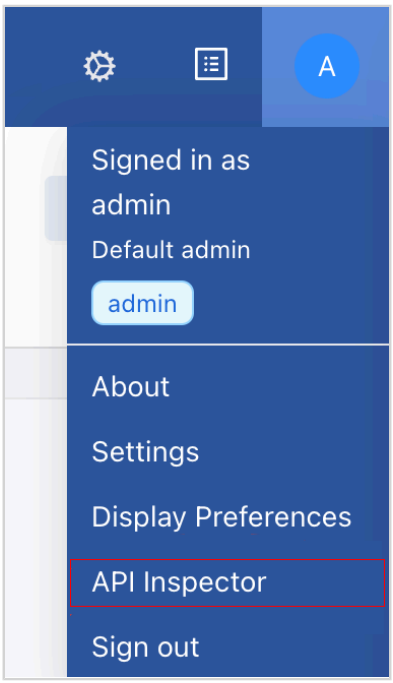
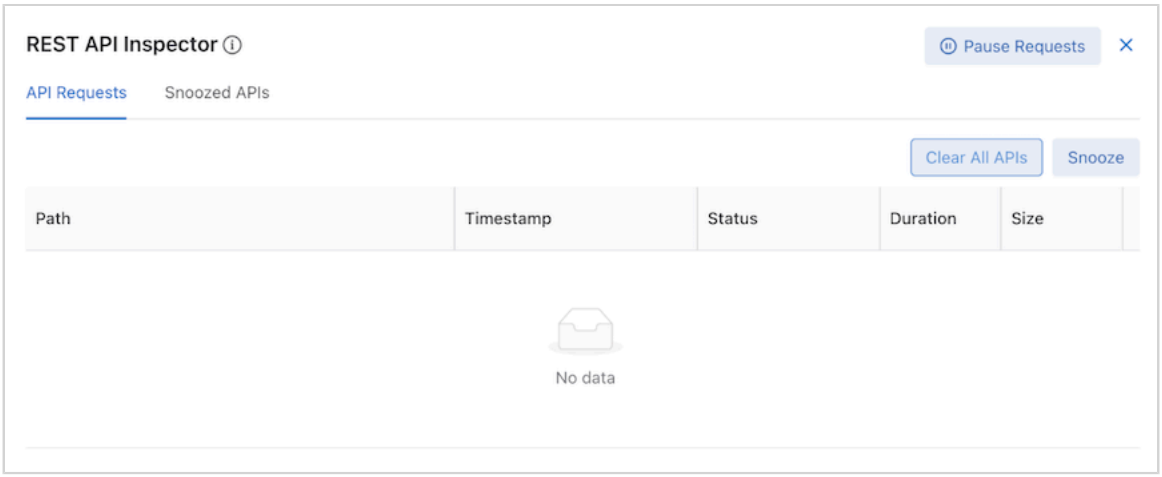


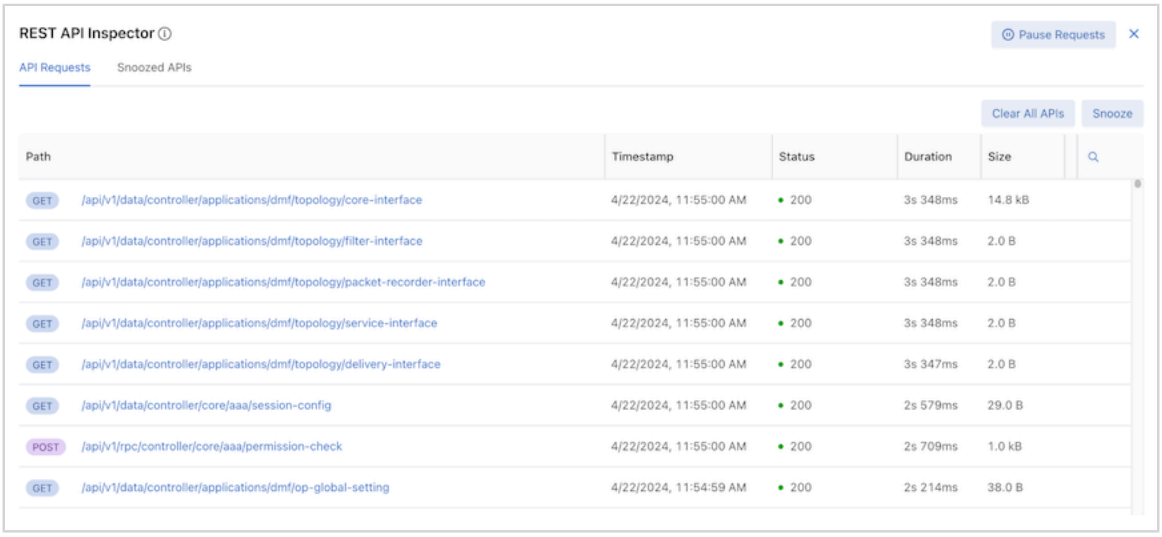
Figure F-2: REST API Inspector



API Requests Tab

API Requests contains a table displaying all recent API requests and several utility buttons.

Figure F-3: API Requests Tab



API Requests Table

Each row contains specific API request information:

- **Path** – HTTP request method and URL
- **Timestamp** – date and approximate time
- **Status** – HTTP response codes
- **Duration** – time taken to receive a response
- **Size** – response body size

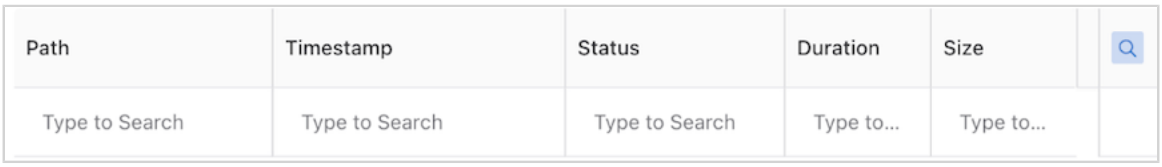
The table logs the latest 500 API requests and automatically updates when new API calls are received.

Search



The table's results are searchable. Selecting the **magnifying glass** icon displays a **Type to Search** line in the first row.

Figure F-4: Search by Column



The table supports search by Path, HTTP Method, and Status column contexts.

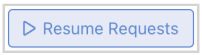
Interaction

Selecting a URL expands specific API requests and enters the **Detailed API Request View**.

Pause Requests Control



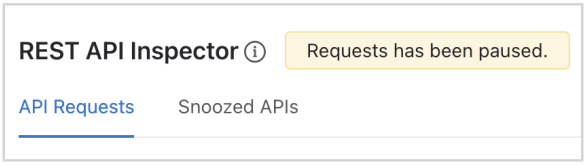
Use **Pause Requests** to stop tracking new API requests. An alert message appears next to the REST API Inspector title.



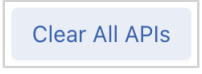
During the pause, the control changes to **Resume Requests**. To resume tracking API requests, select **Resume Requests**. The table updates accordingly.

> **Note:** Any requests made during the pausing state are not retrieved.

Figure F-5: Paused Requests

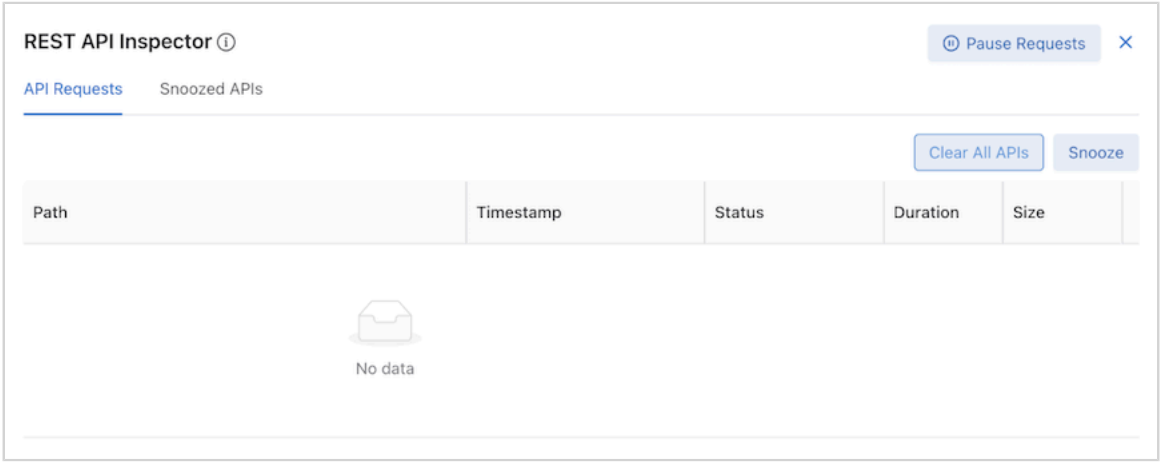


Clear All APIs Button



To clear the entire table, select **Clear All APIs**.

Figure F-6: Cleared Table



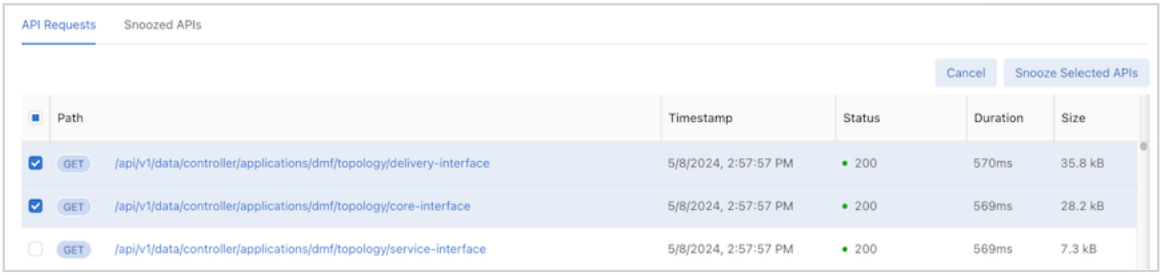
The Snoozed API table is not affected. API Inspector continues receiving and displaying new requests in the table.

Snooze Control



Using **Snooze** makes the table rows selectable, and the Snooze control changes to **Snooze Selected APIs**. Selecting **Cancel** or **Snooze Selected APIs** returns the table entries to an unselected state.

Figure F-7: API Requests



During the selectable state, select specific API Requests and select **Snooze Selected APIs** to prevent them from appearing in the table.

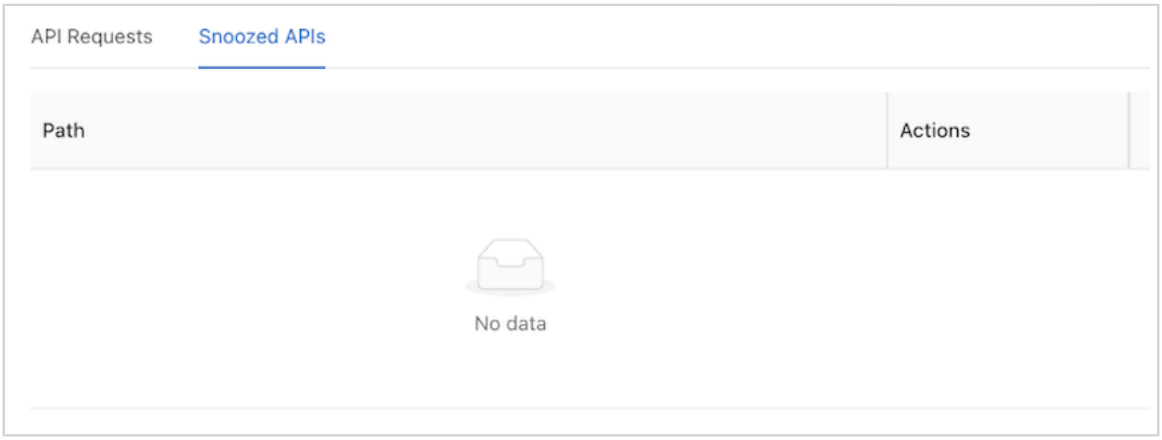
Snoozed APIs Tab

Snoozed APIs display all snoozed API requests.

Snoozed APIs Table

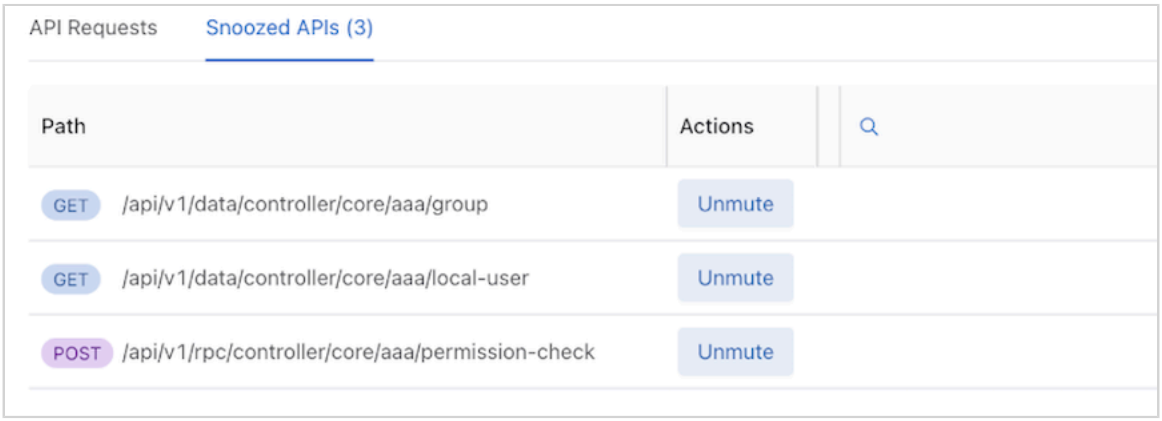
The snoozed APIs table is empty by default.

Figure F-8: Snoozed APIs Default Table



The table displays snoozed APIs. However, unlike the **API Requests** table, the **Snoozed API** table only distinguishes API requests by URL and HTTP Method. Requests with the same URL only appear once in the Snoozed API Table.

Figure F-9: Snoozed APIs



Search



The table's results are searchable. Selecting the **magnifying glass** icon displays a **Type to Search** line in the first row.

Figure F-10: Type to Search



The **Snoozed APIs** table supports search by Path columns contexts only.

Unmute Control

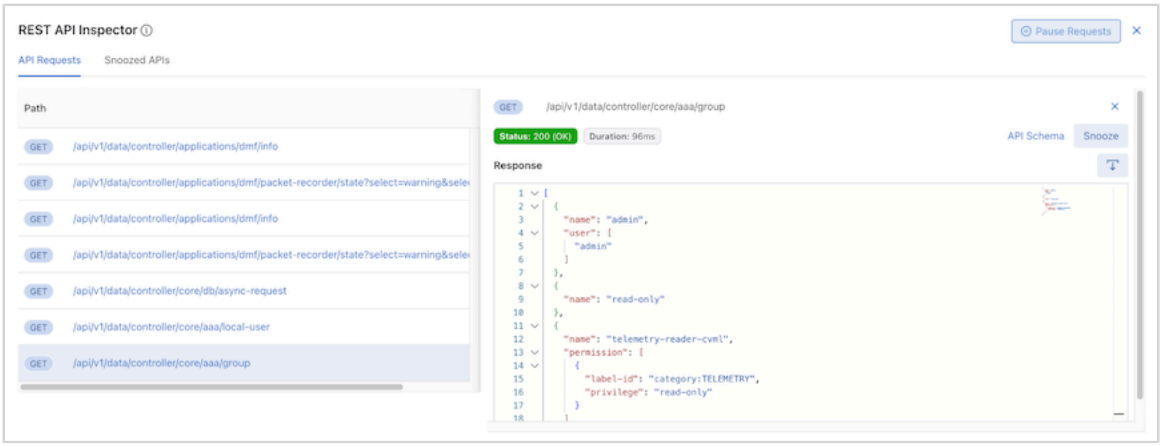


Unmute resumes tracking the API request and updates the table. Any requests with the same URL that occur going forward are added to the API requests table.

Detail API Requests View

Selecting a **URL** in the API Requests table expands that specific API request and enters the **Detail API Requests** view.

Figure F-11: Detailed API Requests View



The UI displays a truncated API Requests table and a detailed view; selecting another **URL** changes the detailed information in the detailed view.

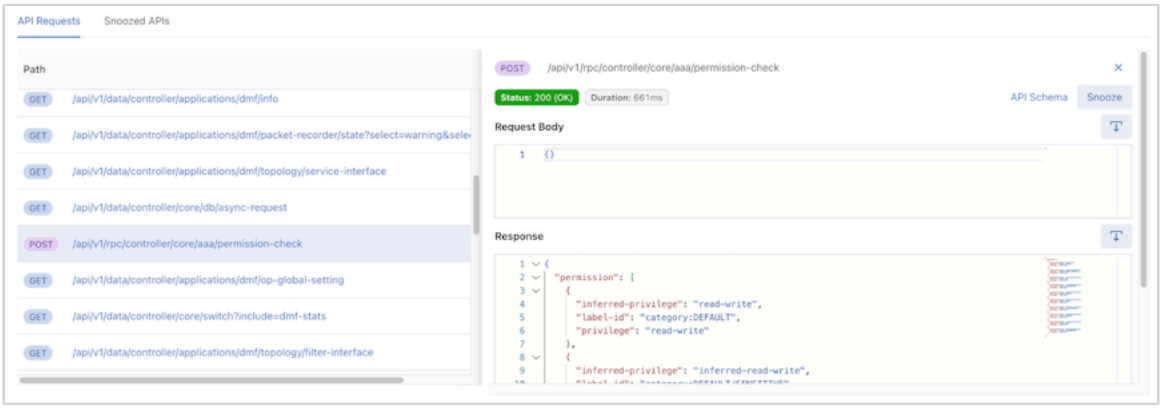
Information on the selected API request includes:

- **HTTP methods** – show the HTTP methods
- **Response status** – show the response status and response text
- **Duration** – show the duration of the API request
- **API Schema** link – Open a new tab to the API Schema Browser Page
- **Snooze** – Snooze this selected API request (same as **Snooze** control)
- **Response** – Show all the response information of this selected API request
- **Download** Icon – Download response information into a .json file.

> **Note:** For PATCH / POST / PUT requests, another field, **Request Body**, shows the request body information.

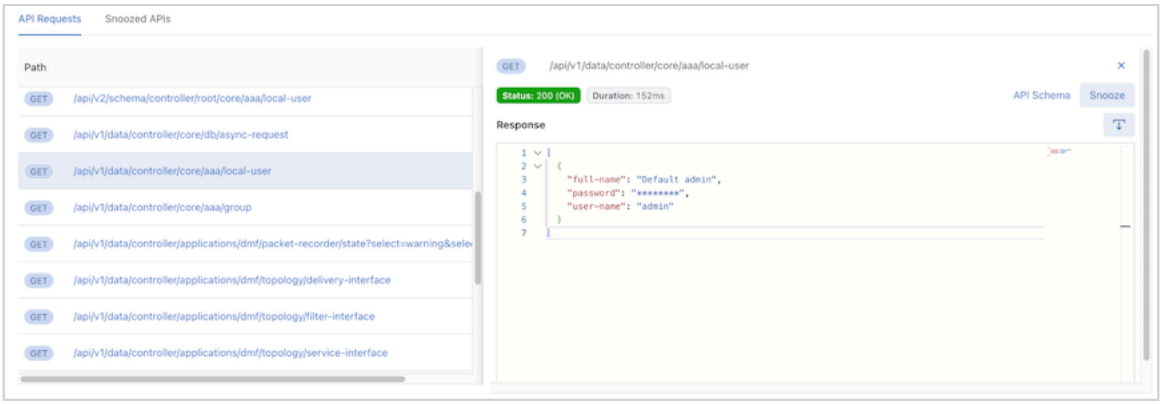
- Use the **download** icon to download the request body information into a .json file.

Figure F-12: Download JSON File



- **API Inspector** obfuscates sensitive values such as password, hash ID, hash password, auth session, encode result, and secret by replacing them with ********.

Figure F-13: Obfuscated Sensitive Data



Appendix G

Configuring Third-Party Services

G.1 Services in the DANZ Monitoring Fabric

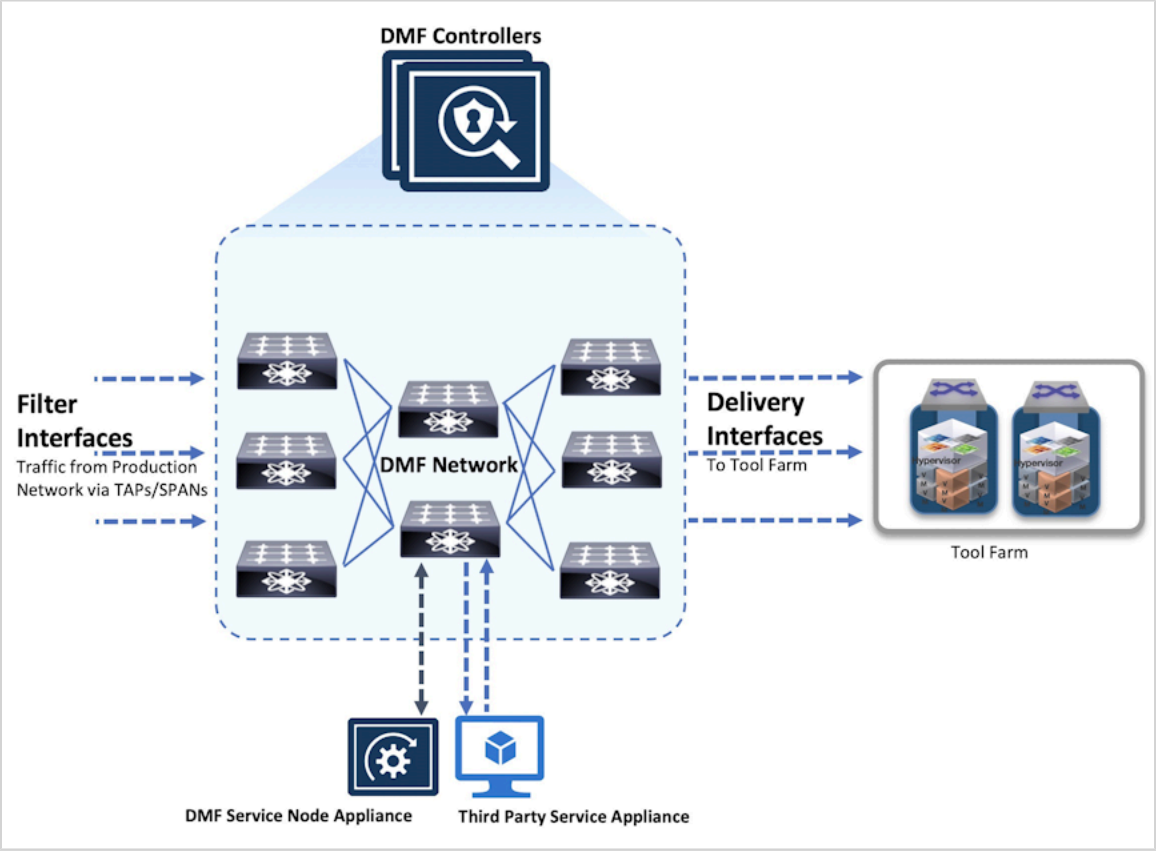
Services in the DANZ Monitoring Fabric (DMF) refer to packet modification operations provided by third-party network packet brokers (NPBs), referred to as service nodes. Services can include operations that refine or modify the data stream delivered to analysis tools.

Each service instance is assigned a numeric identifier because multiple services can be specified for a given policy. Services are applied sequentially, applying a service with a lower sequence number first.

Service nodes are optional devices that process interesting traffic before forwarding it to the delivery ports specified by the policy. Example services include time-stamping packets, packet slicing, or payload obfuscation. To configure a service node:

- Create all the pre-service and post-service interfaces used with the service.
- Use the DMF interface names to create a service node and add pre-service and post-service interfaces.

Figure G-1: Using Services with a Policy



In the figure above, the time-stamping service is applied first, followed by the packet-slicing service. The illustration shows the CLI commands that associate the service with a specific policy. For the illustrated policy, the packet path is as follows:

1. Filter interface (***F3***)

2. Time-stamping service node (pre-service and post-service interfaces)
3. (optional) Packet-slicing service node (pre-service and post-service interfaces)
4. Delivery-interface (***D2***)

Once a policy includes a service, it is only optional if defined explicitly as optional. If not defined as optional in the policy, packet forwarding does not occur when the service is unavailable. For example, configuring the packet-slicing service as optional and a pre-service or post-service interface assigned to that service node is down, the service is skipped, and the packets are delivered to the ***D2*** delivery interface after the time-stamping service is completed. However, if at least one pre-service and post-service interface is unavailable for the time-stamping service, this policy does not forward packets to the delivery interfaces.

Configure all the service interfaces before creating a service definition that uses them.



Note: Before defining a service, first create the service interface names. Otherwise, the service might enter an inconsistent state. If that happens, delete the service definition, create the interfaces, and then re-create the service definition. Alternatively, re-create the service definition without the nonexistent interfaces.

A DMF service can have multiple pre-service and post-service interfaces. Use a Link Access Group (LAG) as a pre-service or a post-service interface.



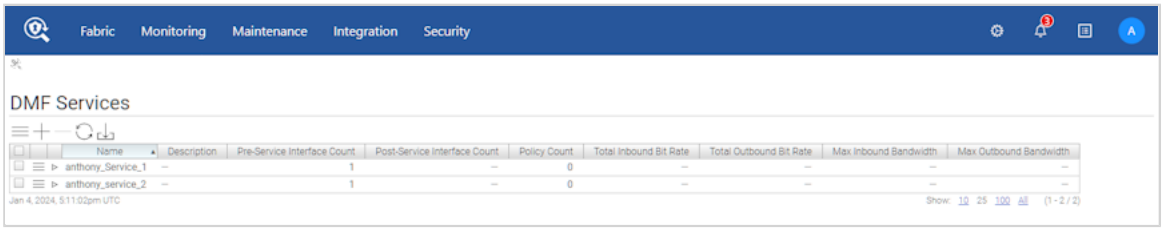
Note: Arista Networks strongly recommends configuring the post-service and pre-service interfaces on the same switch for any DMF service.

G.1.1 Configure a DMF Unmanaged Service

To create a DANZ Monitoring Fabric (DMF) unmanaged service, perform the following steps:

1. Select **Monitoring > Services**.
   The system displays the following table:

   Figure G-2: DMF Unmanaged Service



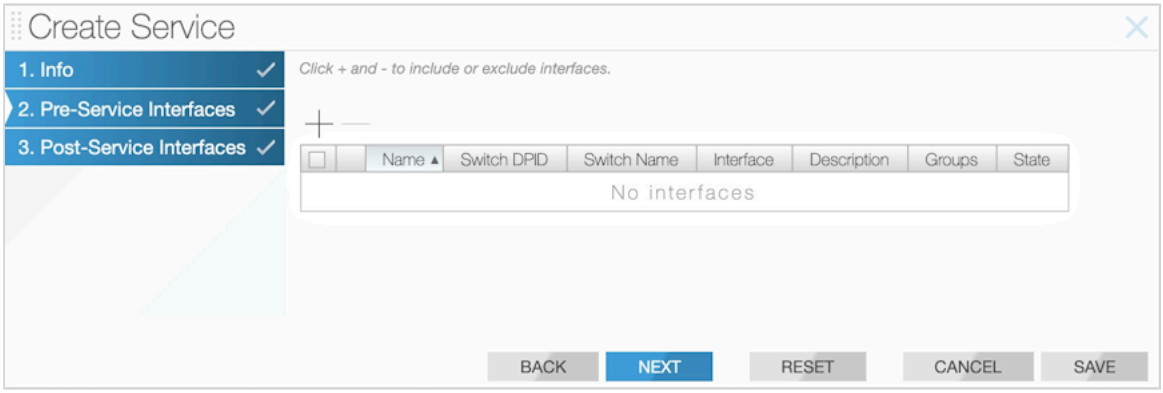
   This table lists the services configured for the DMF. Add, delete, or modify existing services as required.
2. To create a new service, select the provision control (**+**) in the table.
   The system displays the following dialog:

   Figure G-3: Create Service Dialog: Info



3. Type a unique name for the service and optional text description, select **Next**.
   The system displays the following dialog:

   Figure G-4: Create Service Dialog: Pre-service Interfaces

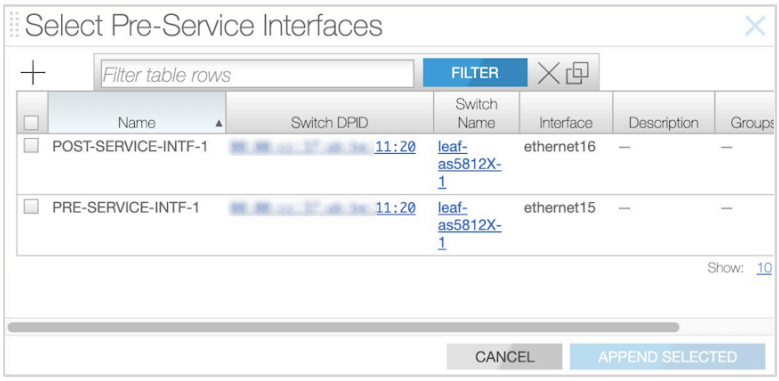


   This table lists the interfaces assigned as pre-service interfaces for the current service.

4. To add a pre-service interface, select the provision control (**+**) in the table.

   The system displays the following dialog:

   Figure G-5: Select Pre-service Interfaces



   This table lists the interfaces available for assignment as pre-service interfaces. To configure a new interface, select the provision control (**+**) in the table. The system displays a dialog for adding a service interface.

5. Enable the checkbox for one or more interfaces to assign as a pre-service interface for the current service and select **Append Selected**.
6. On page two of the **Create Service Interface** dialog, select **Next**.

   The system displays the following dialog:

   Figure G-6: Create Service Dialog: Post-service Interfaces

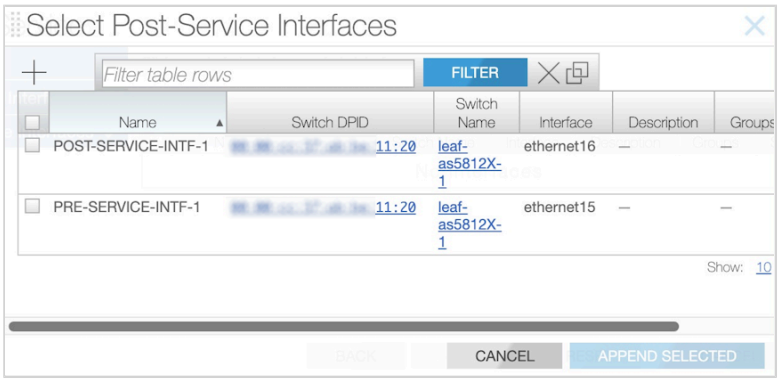


   This table lists the interfaces assigned as post-service interfaces for the current service.

7. To add a post-service interface, select the provision control (**+**) in the table.

   The system displays the following dialog.

   Figure G-7: Select Post-service Interfaces



This table lists the interfaces available for assignment as post-service interfaces.

To configure a new interface, select the provision control (**+**) in the table. The system displays a dialog for adding a service interface, as described in the ***Configuring DMF Unmanaged Services*** section.

8. Enable the checkbox for one or more interfaces to assign as a post-service interface for the current service and select **Append Selected**.
9. Select **Save** on page three of the **Create Service Dialog**.

G.1.2 Configure a DMF Unmanaged Service using the CLI

In the DANZ Monitoring Fabric (DMF), third-party tools that provide packet manipulation services, such as time stamping and packet slicing, are called DMF Unmanaged Services. These optional devices process traffic from filter interfaces before being forwarded to delivery interfaces.



Note: After adding a service to a policy, it is no longer optional unless specifically defining it as optional. If not defined as optional, the policy does not forward packets if the service is unavailable.

To configure an unmanaged service using the CLI, perform the following steps:

1. Create one or more pre-service interfaces for delivering traffic to the NPB, as in the following example.

```
controller-1(config-switch-if)# switch DMF-CORE-SWITCH
controller-1(config-switch-if)# interface s9-eth1
controller-1(config-switch-if)# role service interface-name pre-serv-intf-1
```

2. Create one or more post-service interfaces for receiving traffic from the NPB, as in the following example:

```
controller-1(config-switch-if)# interface s9-eth2
controller-1(config-switch-if)# role service interface-name post-serv-intf-1
```

3. Create a service node and add at least one pre-service and at least one post-service interface using the DMF interface names, as in the following example:

```
controller-1(config)#controller-1(config)# unmanaged-service THIRD-PARTY-SERVICE-1
controller-1(config-unmanaged-srv)# description "this is a third-party unmanaged service"
controller-1(config-unmanaged-srv)# pre-service PRE-SERVICE-INTF-1
controller-1(config-unmanaged-srv)# post-service POST-SERVICE-INTF-1
```

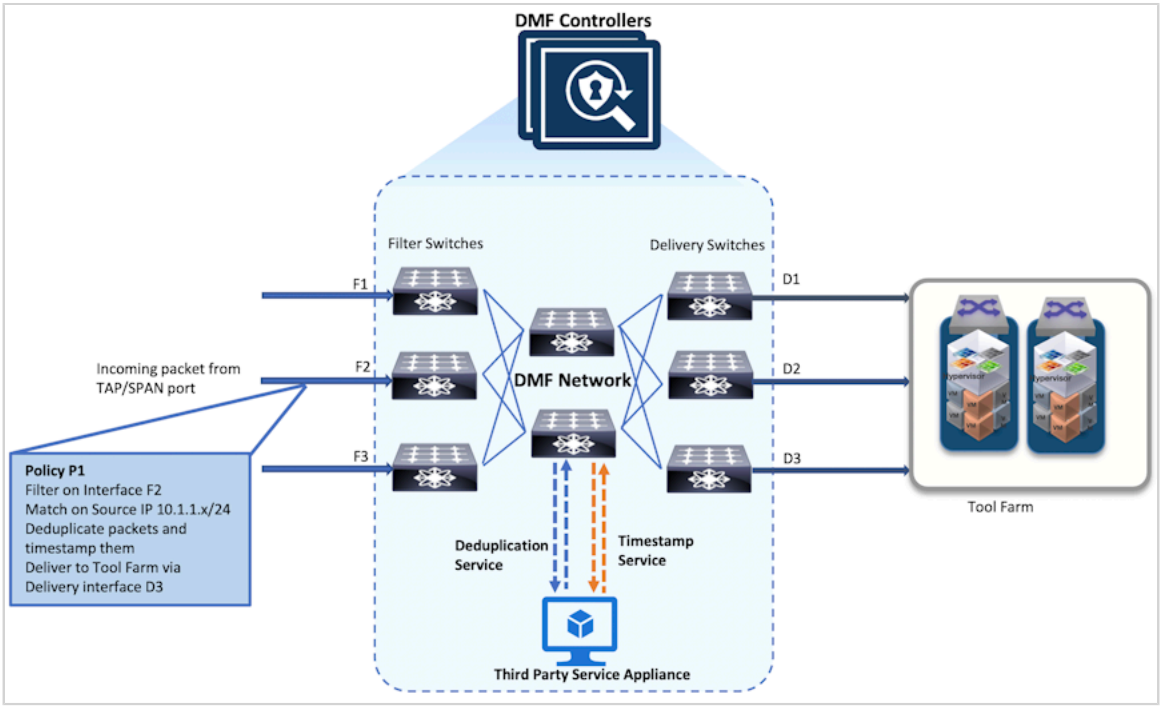
To list the configured services in the DMF fabric, enter the **`show unmanaged-services`** command, as in the following example:

```
controller-1# show unmanaged-service
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Services ~~~~~~~~~~~~~~~~~~~
```

G.2 Service Insertion and Chaining in a DMF Policy

To configure a DANZ Monitoring Fabric (DMF) policy that uses services provided by an NPB, add the **use-service** command to the policy. Services can be configured in series, called chaining, as shown below:

Figure G-8: Service Insertion and Chaining



Because a given policy can specify multiple services, set a sequence number for each service instance so the services are applied in order for the policy traffic. A lower sequence number applies the service first.

To configure a DMF out-of-band policy that uses services provided by an NPB, use the **use-service** command from the config-policy submode to add the service to the policy.

The following are the configuration commands for implementing the illustrated example:

```
controller-1(config)# policy DMF-POLICY-1
controller-1(config-policy)# use-service UMS-DEDUPLICATE-1 sequence 100
controller-1(config-policy)# use-service UMS-TIMESTAMP-1 sequence 101
```

In this example, the packet deduplication service is applied first, followed by time stamping. If all the pre-service or post-service interfaces for the packet-slicing service nodes are down, then this service is skipped if configured as optional. In this example, the time-stamping service is applied before the packet deduplication service, and the packet deduplication service is configured as optional.

```
controller-1(config)# policy DMF-POLICY-1
controller-1(config-policy)# use-service UMS-TIMESTAMP-1 sequence 100
controller-1(config-policy)# use-service UMS-DEDUPLICATE-1 sequence 101 optional
.. note::
If a service is inserted, the policy can only become active and begin forwarding when at
least one delivery port is reachable from all the post-service interfaces defined for the service.
```

Enter the **show policy** command from any mode to display the run time services being applied.

Appendix H

References

H.1 Related Documents

The following documentation is available for ***DANZ Monitoring Fabric***:

- *DANZ Monitoring Fabric Release Notes*
- *DANZ Monitoring Fabric User Guide*
- *DANZ Monitoring Fabric Deployment Guide*
- *DANZ Monitoring Fabric Hardware Compatibility List*
- *DANZ Monitoring Fabric Hardware Guide*
- *DANZ Monitoring Fabric Verified Scale Guide*
- *Arista Analytics User Guide*
- *DANZ Monitoring Fabric REST API Guide*
- *DANZ Monitoring Fabric SNMP MIB Reference Guide*
- *DANZ Monitoring Fabric Log Message Guide*